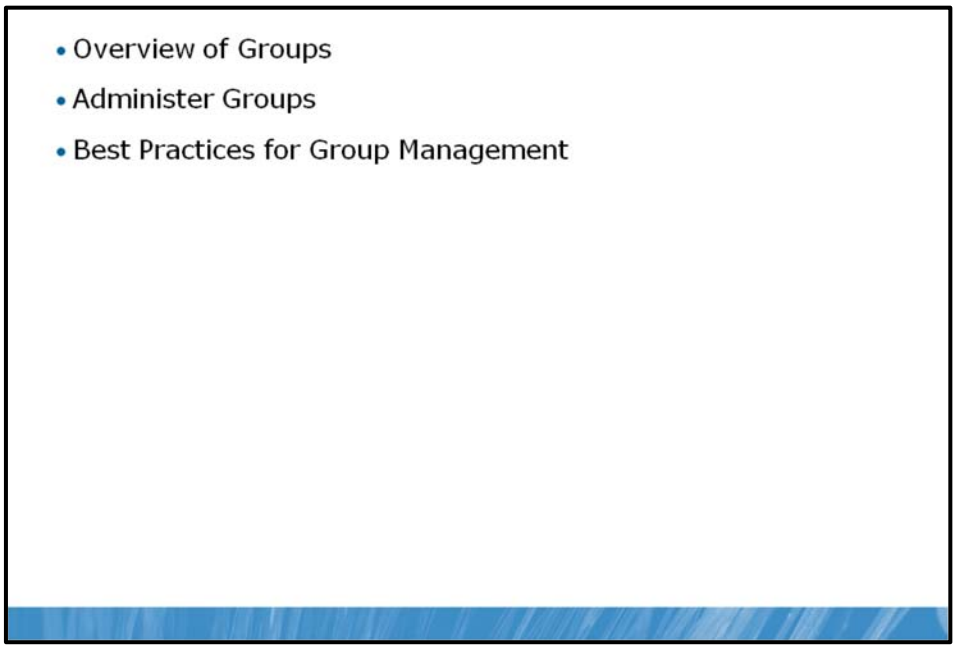
Module 4

Managing Groups

Contents:

Module Overview



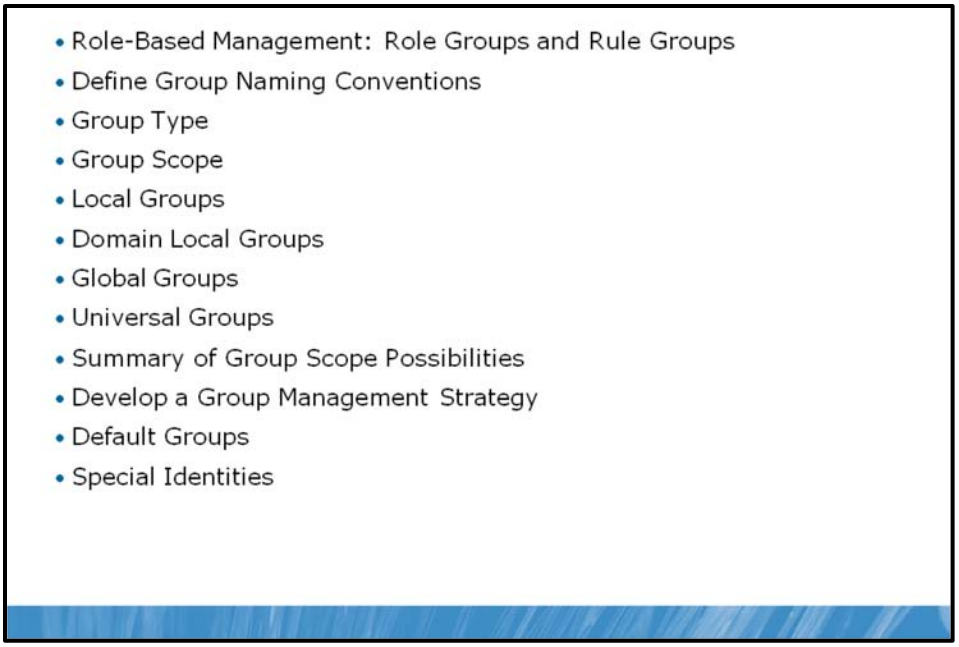
Although users and computers, and even services, change over time, business roles and rules tend to remain more stable. Your business probably has a finance role, which requires certain capabilities in the enterprise. The user or users who perform that role will change, but the role will remain. For that reason, it is not practical to manage an enterprise by assigning rights and permissions to individual users, computers, or service identities. Management tasks should be associated with groups. In this course, you will use groups to identify administrative and user roles, to filter Group Policy, to assign unique password policies, to assign rights and permissions, and more. To prepare for those tasks, in this module, you will learn how to create, modify, delete, and support group objects in an Active Directory® domain.

Objectives

After completing this module, you will be able to:

- Describe the role of groups in managing an enterprise.
- Administer groups by using the built-in tools in Windows Server 2008.
- Describe the best practices for managing groups.

Lesson 1
Overview of Groups



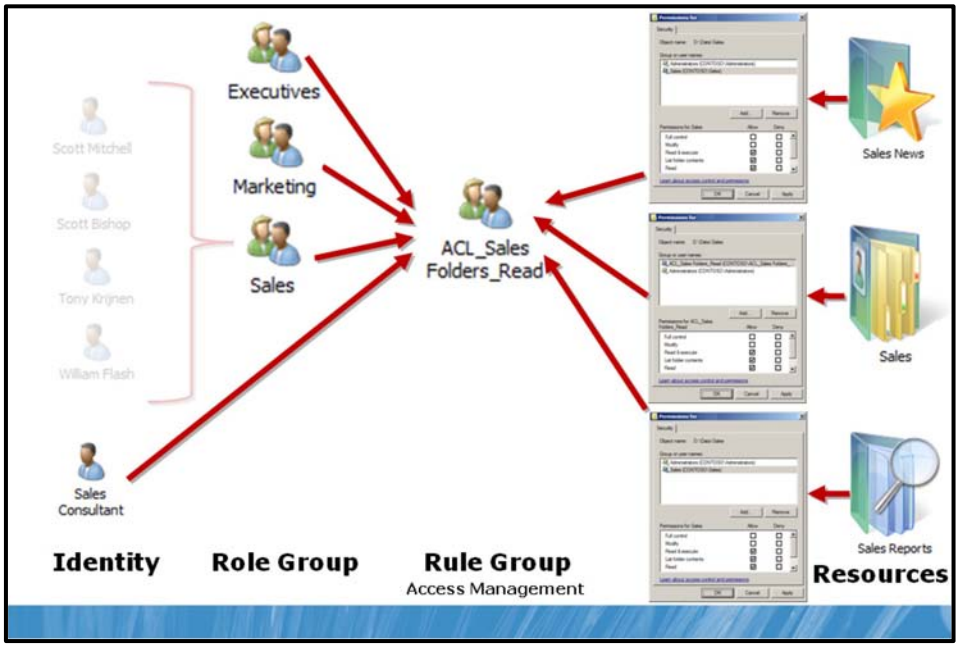
Before implementing groups in your environment, you should learn and understand how groups are used and which types of groups exist. It is important to understand group scope so that you can identify proper group type in various scenarios. Also, it is very important to define proper group naming convention and to understand how groups can be nested in other groups and the benefits of that approach.

Objectives

After completing this lesson, you will be able to:

- Understand the role of groups in managing an enterprise.
- Define group naming conventions.
- Understand group types.
- Understand group scope.
- Identify group membership and nesting possibilities.
- Understand how to manage and administer groups
- Understand the best practice for group nesting to achieve role-based management.

Role-Based Management: Role Groups and Rule Groups



Imagine next that it is not only sales people who require Read access to the folders. Executives, Marketing department employees, and the sales consultant hired by your organization also require Read permission to the same folders. It is very common that various groups of users require access to same resources.

You could add those groups to the ACL of the folders, granting each of them Allow Read permission, but soon you will end up with an ACL with multiple permissions, this time assigning the Allow Read permission to multiple groups, instead of multiple users. To give the three groups and one user permission to the three folders on the three servers, you will have to add twelve permissions! The next group that requires access will require three more changes to grant permissions to the ACLs of the three shared folders.

What if eight users who are not sales people, marketing employees, or executives, have a business need for Read access to the three folders? Do you add their individual user accounts to the ACLs? If so, that is 24 more permissions to add and manage!

You can see that using only one type of group—a role group that defines the business roles of users—quickly becomes an ineffective way of enabling management of access to the three folders. If the management rule suggests that three roles and nine additional users require access to the resource, you are assigning a total of 36 permissions on ACLs. It becomes very difficult to maintain compliance and to audit. Even simple questions such as, "Can you tell me every user who can read the Sales folders?" become difficult to answer.

The solution is to recognize that there are two types of management that must take place to effectively manage this scenario. You must manage the users as collections, based upon their business roles; and, separately, you must manage access to the three folders.

The three folders are also a collection of items. They are a single resource—a collection of Sales folders—that just happens to be distributed across three folders on three servers. You are trying to manage Read access to that resource. You need a single point of management with which to manage access to the resource.

This requires another group—a group that represents Read access to the three folders on the three servers. We call that type of group a rule group (sometimes, also resource groups). Imagine that you create a group called ACL_Sales Folders_Read. This group will be assigned the Allow Read permission on the three folders. The Sales, Marketing, and Executives groups, along with the individual users, will all be members of the ACL_Sales Folders_Read group. You assign only three permissions: one on each folder, granting Read access to the ACL_Sales Folders_Read group.

The ACL_Sales Folders_Read group becomes the focus of access management. As additional groups or users require access to the folders, they will be added to that group. It also becomes easier to report who has access to the folders. Instead of having to examine the ACLs on each of the ten folders, you simply examine the membership of the ACL_Sales Folders_Read group.

To effectively manage even a slightly complex enterprise, you will need two "types" of groups that perform two distinct purposes:

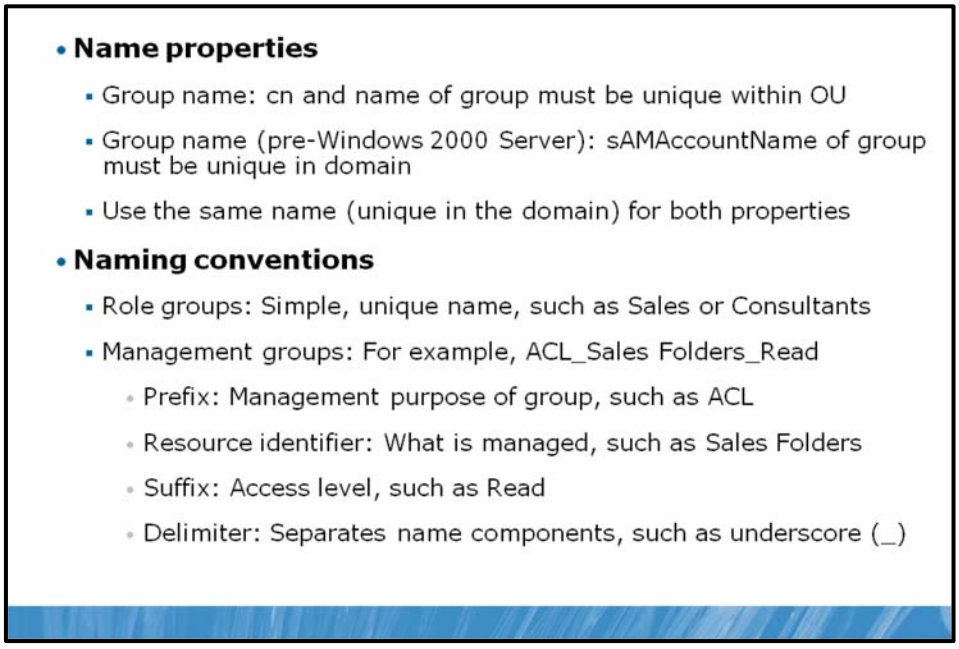
- Groups that define roles. These groups, referred to as *role groups*, contain users, computers, and other role groups based on common business characteristics such as location, job type, and so on.
- Groups that define management rules. These groups, referred to as *rule groups*, define how an enterprise resource is being managed.

This approach to managing the enterprise with groups is called *role-based management*. You define roles of users based on business characteristics—for example, department or division affiliation such as sales, marketing, and executives, and you define management rules—for example, the rule that manages which roles and individuals can access the three folders.

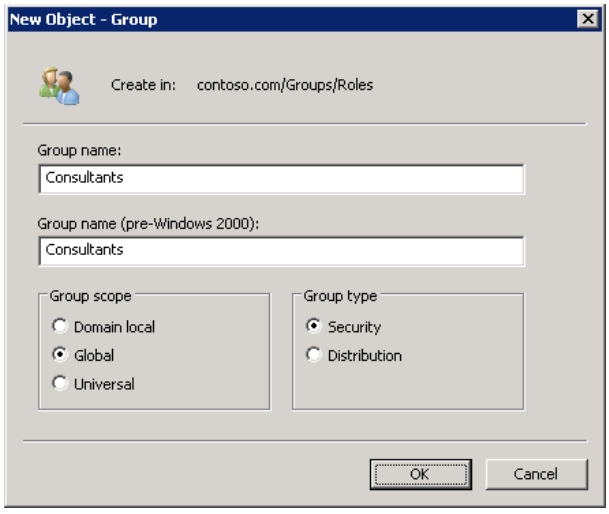
You can achieve both management tasks by using groups in a directory. Roles are represented by groups that contain users, computers, and other roles. Roles can include other roles, for example, a Manager role might include Sales Managers, Finance Managers, and Production Managers roles. Management rules, such as the rule that defines and manages Read access to the three folders, are represented by groups also. Rule groups contain roles, and occasionally, individual users or computers such as the sales consultant and eight other users in the example.

The key takeaway is that there are two **"**types**"** of groups: one that defines the role, and the other that defines how a resource is managed.

Define Group Naming Conventions



To create a group by using the Active Directory Users and Computers snap-in, you should right-click the organizational unit (OU) in which you want to create a group, point to New, and then click Group. The **New Object - Group** dialog box, shown in the following image, allows you to specify fundamental properties of the new group.



The following name properties can be configured in this dialog box:

- **Group name.** The cn and name of group object must be unique only within the OU
- **Group name (pre-Windows 2000).** *sAMAccountName* of group, unique in domain

Important best practice Use the same name (unique in the domain) for both properties.

The first property you must configure are the group's names. A group, like a user or computer, has several names. The first, shown in the Group Name box above, is used by Windows 2000 and later systems to identify the object—it becomes the cn, and name attributes of the object. The second, the pre-Windows 2000 name, is the sAMAccountName attribute, used to identify the group to computers running Windows NT 4.0 and to some devices, such as network attached storage (NAS) devices running non-Microsoft operating systems. The cn and name attributes must be unique only within the container—the OU—in which the group exists. The sAMAccountName must be unique in the entire domain. Technically, the sAMAccountName could be a different value than the cn and name, but it is highly discouraged to make these different. Pick a name that is unique in the domain, and use it in both name fields in the **New Object – Group** dialog box.

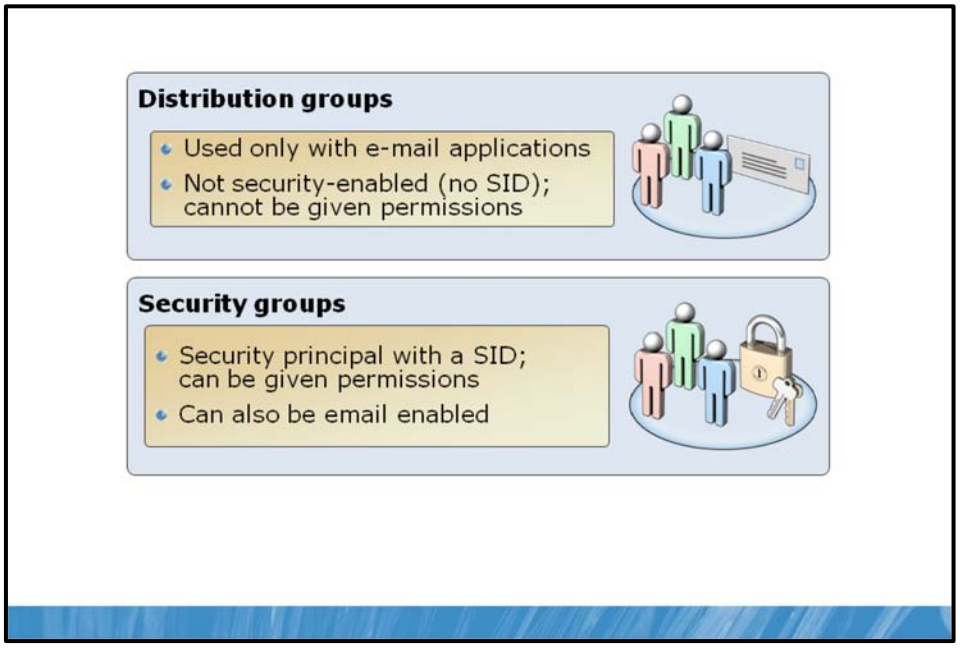
The name you choose should help you manage the group and manage your enterprise on a day-to-day basis. We recommend that you follow a naming convention that identifies the type of group and the purpose of the group.

- Role groups. Simple, unique name, such as Sales or Consultants
- Management groups. For example, ACL_Sales Folders_Read

- Prefix. This identifies the management purpose of group, such as ACL for groups managing access permissions to shared resources. It is used on access control lists, so the prefix ACL is used.
- Resource identifier. This is a unique identifier for what is being managed. The main part of the name uniquely identifies the resource that is being managed with the group, in this example, Sales Folders.
- Suffix. The suffix further defines what is being managed by the group. In the case of resource access management groups, the suffix defines the level of access provided to members of the group. In our example, that is Read.
- Delimiter. This should be a consistently used marker separating prefix, identifier, and suffix, such as an underscore (_). Do not use the delimiter elsewhere in the name—use it only as a delimiter. Note that the delimiter is not used between the words Sales and Folder. Spaces are acceptable in group names—you will just need to enclose group names in quotes when you refer to them in commands or in scripts. You can create scripts that use the delimiter to deconstruct group names to facilitate auditing and reporting**.**

Keep in mind that role groups that define user roles will often be used by non-technical users. For example, you might email enable the Sales group so that it can be used as an email distribution list. Therefore, we recommend that you keep your naming convention for role groups simple and straightforward. In other words, your naming convention for role groups is *not* to use prefixes or suffixes or delimiters—just a user-friendly, descriptive name.

Group Type



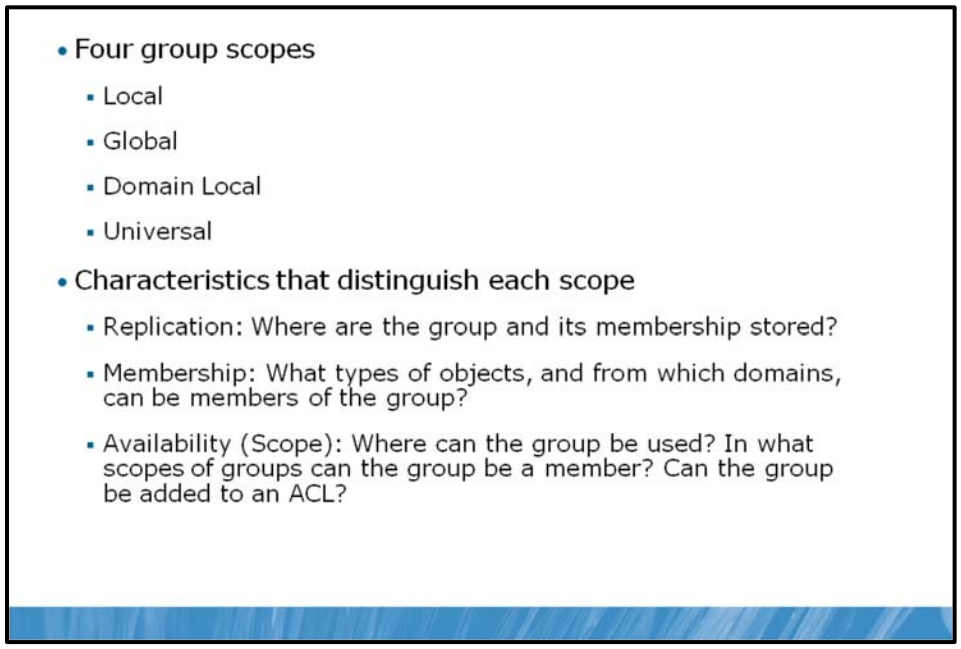
There are two types of groups: security and distribution. When you create a group, you make the selection of the group type in the **New Object – Group dialog** box.

Distribution groups are used primarily by email applications. These groups are not security enabled—they do not have SIDs—so they cannot be given permission to resources. Sending a message to a distribution group sends the message to all members of the group.

Security groups are security principals with SIDs. These groups can therefore be used in permission entries in ACLs to control security for resource access. Security groups can also be used as distribution groups by email applications. If a group will be used to manage security, it must be a security group.

Because security groups can be used for both resource access and email distribution, many organizations use only security groups. However, we recommend that if a group will be used only for email distribution, you should create the group as a distribution group. Otherwise, the group is assigned a SID, and the SID is added to the user's security access token, which can lead to unnecessary size increase of the security token.

Group Scope



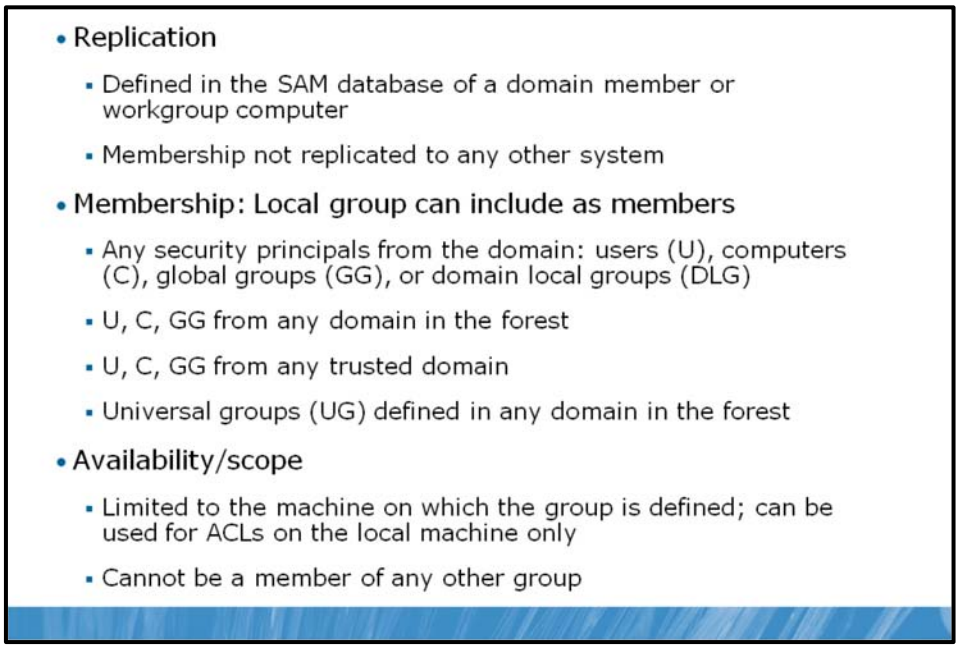
Groups have members: users, computer, and other groups; groups can be members of other groups; and groups can be referred to by ACLs, Group Policy object (GPO) filters, and other management components. Group scope impacts each of these characteristics of a group: what it can contain, what it can belong to, and where it can be used. There are four group scopes: global, domain local, local, and universal.

The characteristics that define each scope fall into these categories:

- Replication. Where is the group defined, and to what systems is the group replicated?
- Membership. What types of security principals can the group contain as members? Can the group include security principals from trusted domains?
- In Module 14, you will learn about trust relationships, or trusts. A trust allows a domain to refer to another domain for user authentication, to include security principals from the other domain as group members, and to assign permissions to security principals in the other domain. The terminology used can be confusing. If Domain A trusts Domain B, Domain A is the trusting domain and Domain B is the trusted domain. Domain A accepts the credentials of users in Domain B. It forwards requests by Domain B users to authenticate to a domain controller in Domain B, because it trusts the identity store and authentication service of Domain B. Domain A can add Domain B's security principals to groups and ACLs in Domain A.
- Availability. Where can the group be used? Is the group available to add to another group? Is the group available to add to an ACL?

Keep these broad characteristics in mind as you explore the details of each group scope.

Local Groups



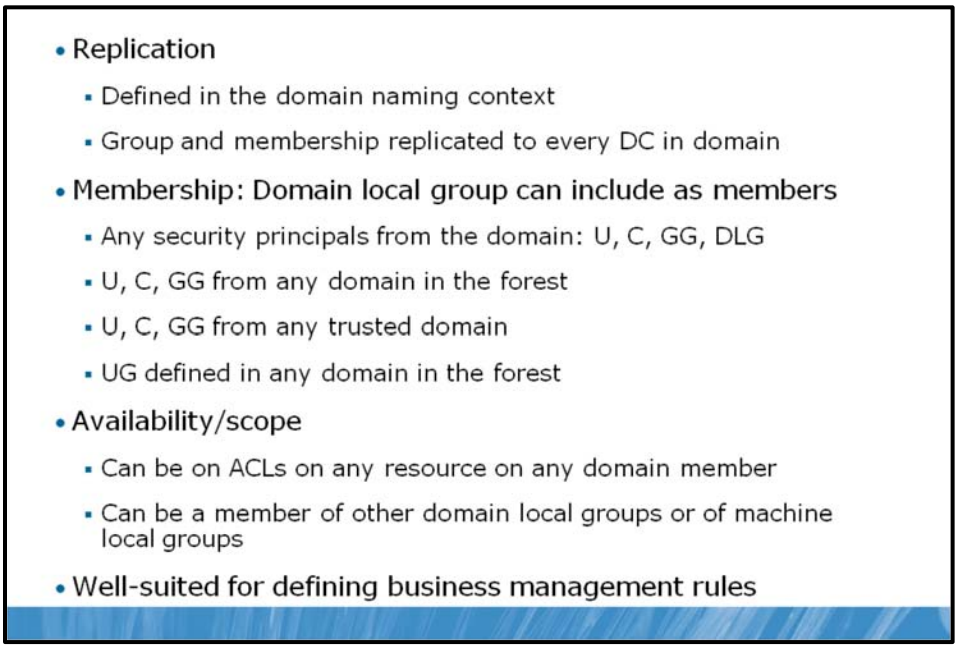
Local groups are truly local—defined on and available to a single computer. Local groups are created in the security accounts manager (SAM) database of a domain member computer—both workstations and servers have local groups. Local groups have the following characteristics:

- Replication. A local group is defined only in the local SAM database of a domain member. The group and its membership are not replicated to any other system.
- Membership. A local group can include as members:
- Any security principals from the domain—users, computers, global groups, or domain local groups.
- Users, computers, and global groups from any domain in the forest.
- Users, computers, and global groups from any trusted domain.
- Universal groups defined in any domain in the forest.
- Availability. A local group has only machine-wide scope. It can be used in ACLs on the local machine only. A local group cannot be a member of any other group.

Best Practice

In a workgroup, you use local groups to manage security of resources on a system. In a domain, however, managing the local groups of individual machines becomes unwieldy, and is for the most part unnecessary. We do not recommend creating custom local groups on domain members. There are very few scenarios in a domain environment that are addressed by using local groups. In most cases, the Users and Administrators local groups are the only local groups that you should be concerned with managing, in a domain environment.

Domain Local Groups



Domain local groups are used primarily to manage permissions to resources, which means they mostly serve as rule groups. For example, the ACL_Sales Folders_Read group discussed earlier in the lesson would be created as a domain local group. Domain local groups have the following characteristics:

- Replication. A domain local group is defined in the domain naming context. The group object and its membership (the member attribute) are replicated to every domain controller in the domain.
- Membership. A domain local group can include as members:
  - Any security principals from the domain—users, computers, global groups, or other domain local groups.
  - Users, computers, and global groups from any domain in the forest.
  - Users, computers, and global groups from any trusted domain.
  - Universal groups defined in any domain in the forest.
- Availability. A domain local group can be added to ACLs on any resource on any domain member. Additionally, a domain local group can be a member of other domain local groups, or even machine local groups.

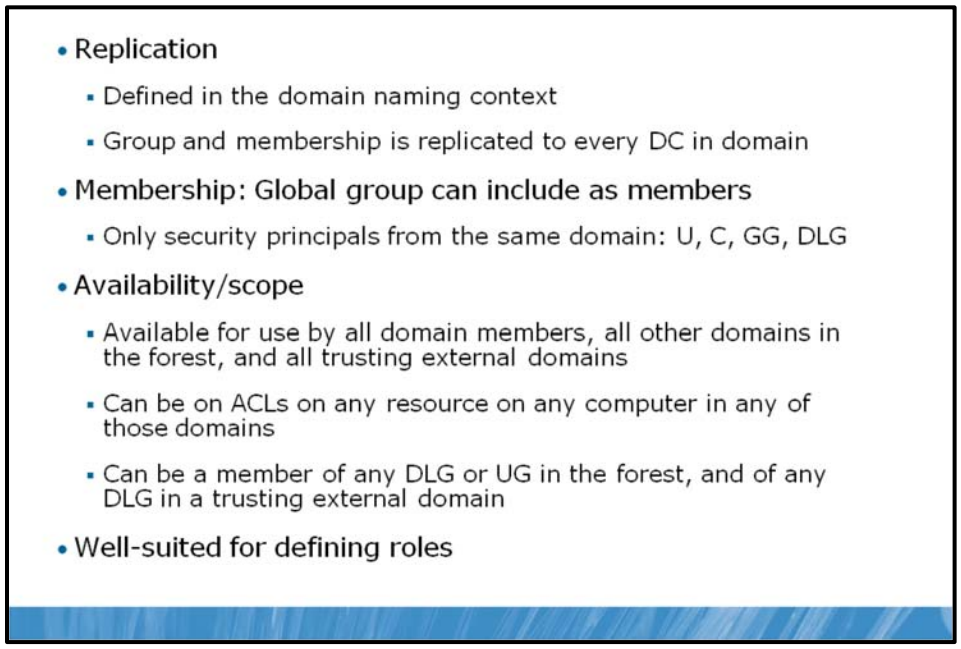
The membership capabilities of a domain local group (the groups to which a domain local group can belong) are identical to those of local groups, but the replication and availability of the domain local group make it useful across the entire domain.

Best Practice

Domain local groups are well suited for defining business management rules, such as resource access rules, because the group can be applied anywhere in the domain, and it can include members of any type within the domain, and members from trusted domains.

For example, a domain local security group named ACL_Sales Folders_Read might be used to manage Read access to a collection of folders that contain sales information on one or more servers.

Global Groups



Global groups are used primarily to define collections of domain objects based on business roles, which means that they mostly serve as role groups. Role groups, such as the Sales and Marketing groups mentioned earlier, and roles of computers such as a Sales Laptops group, are created as global groups. Global groups have the following characteristics:

- Replication. A global group is defined in the domain naming context. The group object, including the member attribute, is replicated to all domain controllers in the domain.
- Membership. A global group can include as members only those users, computers, and other global groups in the same domain.
- Availability. A global group is available for use by all domain members, and by all other domains in the forest and all trusting external domains. A global group can be a member of any domain local or universal group in the domain or in the forest. It can also be a member of any domain local group in a trusting domain. Finally, a global group can be added to ACLs in the domain, in the forest, or in trusting domains.

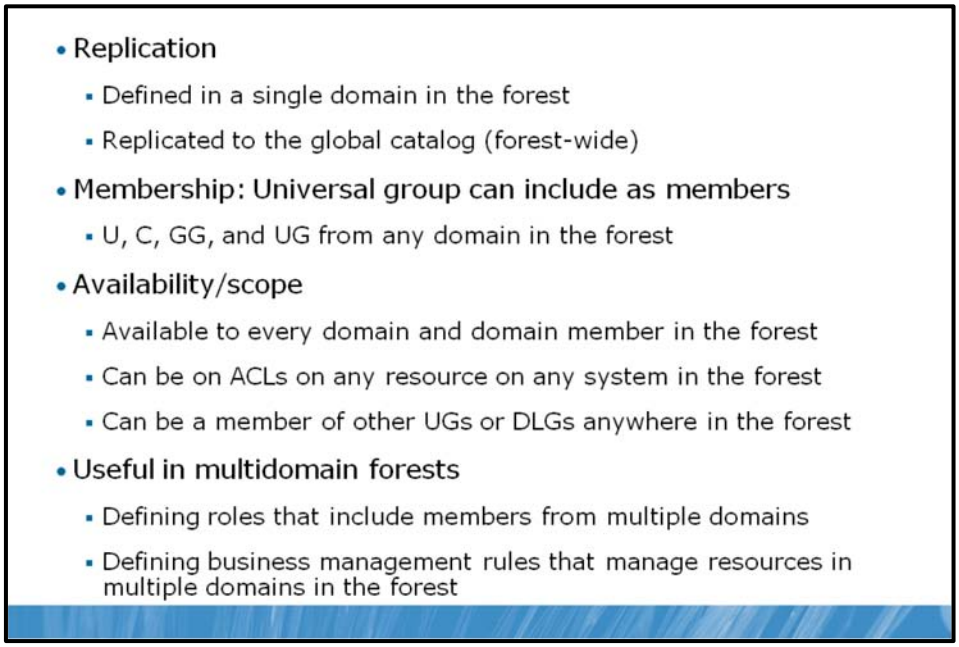
As you can see, global groups have the most limited membership (only users, computers, and global groups from the same domain) but the broadest availability across the domain, the forest, and trusting domains.

Best Practice

Global groups are well suited to defining roles, because roles are generally collections of objects from the same directory.

For example, global security groups named Consultants and Sales might be used to define users who are consultants and sales people, respectively.

Universal Groups



Unlike Global and Domain local groups, the use of Universal Groups is not limited to role or rule type of groups; they can be used in both types of groups depending on the scenario.

Universal groups have the following characteristics:

- Replication. A universal group is defined in a single domain in the forest but is replicated to the global catalog. You will learn more about the global catalog in Module 12. Objects in the global catalog will be readily accessible across the forest.
- Membership. A universal group can include as members users, global groups, and other universal groups from any domain in the forest.
- Availability. A universal group can be a member of a universal group or domain local group anywhere in the forest. Additionally, a universal group can be used to manage resources, for example, to assign permissions, anywhere in the forest.

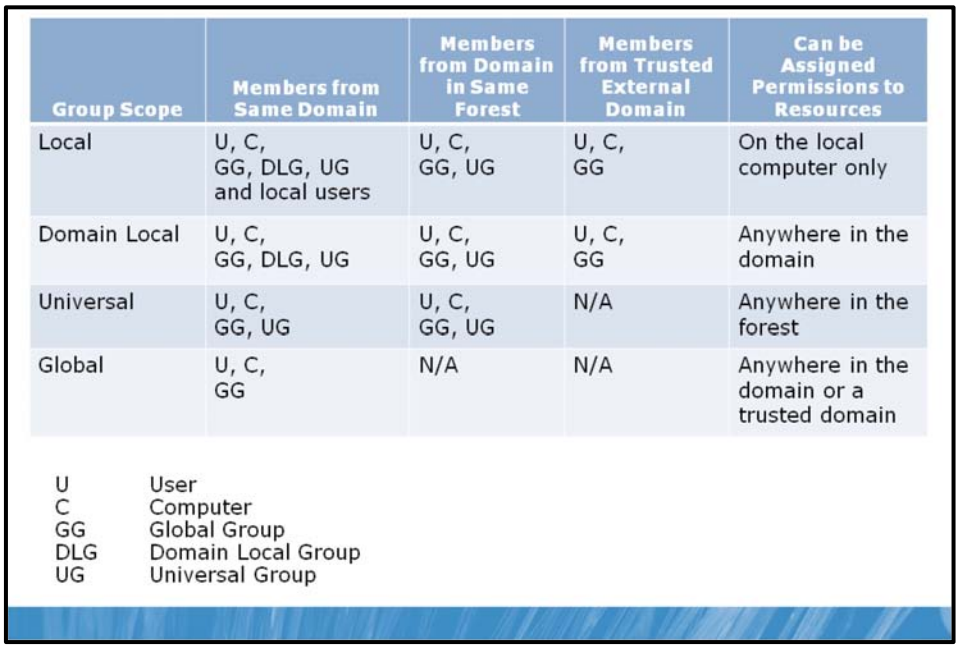
Universal groups are useful in multidomain forests. They allow you to define roles or to manage resources that span more than one domain. The best way to understand universal groups is through an example: Trey Research has a forest with three domains: Americas, Asia, and Europe. Each domain has user accounts and a global group called, Regional Managers, which includes the managers of that region. Remember that global groups can contain only users from the same domain. A universal group called, Trey Research Regional Managers, is created, and the three Regional Managers groups are added as members. The Trey Research Regional Managers group therefore defines a role for the entire forest. As users are added to any one of the Regional Managers groups, they will, through group nesting, be members of the Trey Research Regional Managers.

Trey Research is planning to release a new product that requires collaboration across its regions. Resources related to the project are stored on file servers in each domain. To define who has the ability to modify files related to the new product, a universal group is created called ACL_New Product_Modify. That group is assigned the Allow Modify permission to the shared folders on each of the file servers in

each of the domains. The Trey Research Regional Managers group is made a member of the ACL_New Product_Modify group, as are various global groups and a handful of users from each of the regions.

As you can see from this example, universal groups can help you to represent and consolidate roles that span domains in a forest, and to define rules that can be applied across the forest.

Summary of Group Scope Possibilities



Group Scope	Members from Same Domain	Members from Domain in Same Forest	Members from Trusted External Domain	Can be Assigned Permissions to Resources
Local	U, C, GG, DLG, UG and local users	U, C, GG, UG	U, C, GG	On the local computer only
Domain Local	U, C, GG, DLG, UG	U, C, GG, UG	U, C, GG	Anywhere in the domain
Universal	U, C, GG, UG	U, C, GG, UG	N/A	Anywhere in the forest
Global	U, C, GG	N/A	N/A	Anywhere in the domain or a trusted domain

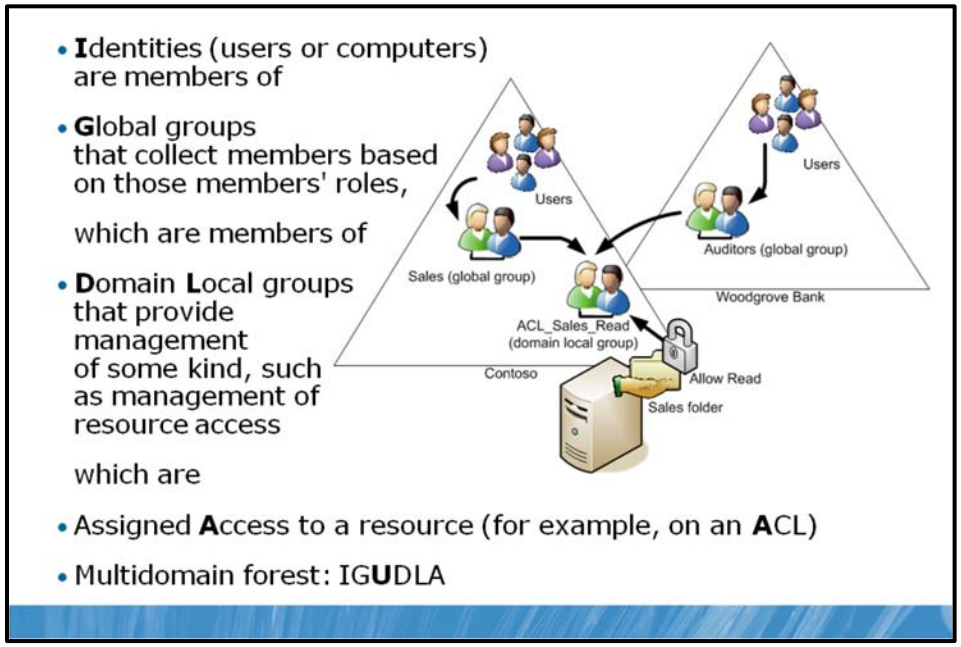
U User
C Computer
GG Global Group
DLG Domain Local Group
UG Universal Group

In day-to-day administration, it is important that you be completely familiar with the membership characteristics of each group scope.

The following table summarizes the objects that can be members of each group scope.

Group Scope	Members from the Same Domain	Members from Another Domain in the Same Forest	Members from a Trusted External Domain
Local	• Users • Computers • Global groups • Universal groups • Domain local groups • Also, local users defined on the same computer as the local group	• Users • Computers • Global groups • Universal groups	• Users • Computers • Global groups
Domain Local	• Users • Computers • Global groups • Domain local groups • Universal groups	• Users • Computers • Global groups • Universal groups	• Users • Computers • Global groups
Universal	• Users • Computers • Global groups • Universal groups	• Users • Computers • Global groups • Universal groups	N/A
Global	• Users • Global groups	N/A	N/A

Develop a Group Management Strategy



Adding groups to other groups—a process called nesting—can create a hierarchy of groups that support your business roles and management rules. Now that you have learned the business purposes and technical characteristics of groups, it is time to align the two in a strategy for group management.

Earlier in this lesson, you learned what types of objects can be members of each group scope. Now it is time to identify what types of objects should be members of each group scope. This leads to the best practice for group nesting, known as IGDLA. IGDLA stands for Identities, Global groups, Domain local groups, and Access:

- ***I***dentities (user and computer accounts) are members of:
- ***G***lobal groups that represent business roles. Those role groups (global groups) are members of:
- ***D***omain Local groups that represent management rules—determining who has Read permission to a specific collection of folders, for example. These rule groups (domain local groups) are granted:
- ***A***ccess to resources. In the case of a shared folder, access is granted by adding the domain local group to the folder's access control list (ACL), with a permission that provides the appropriate level of access.

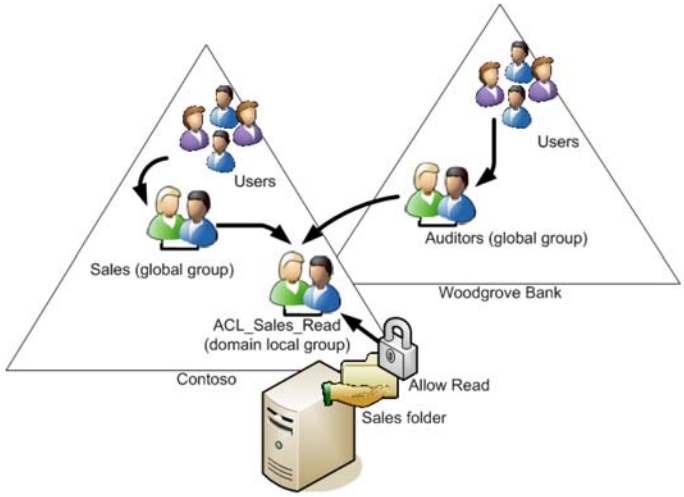
Note This approach of groups nesting was earlier known as AGDLP, that is, Accounts, Global Groups, Domain Local Groups, Permissions. However, the terminology used in this course, IGDLA, has more general scope of appliance and it also aligns with industry-standard terminology.

In a multidomain forest, there are universal groups also, which fit in between global and domain local groups. Global groups from multiple domains are members of a single universal group. That universal

group is a member of domain local groups in multiple domains. You can remember the nesting as *IGUDLA*.

IGDLP Example

This best practice for implementing group nesting translates well even in multidomain scenarios. Consider the figure below, which describes usage of IGDLP scenario:



This figure represents a group implementation that reflects not only the technical view of group management best practices (IGDLA), but also the business view of role-based, rule-based management.

Consider the following scenario:

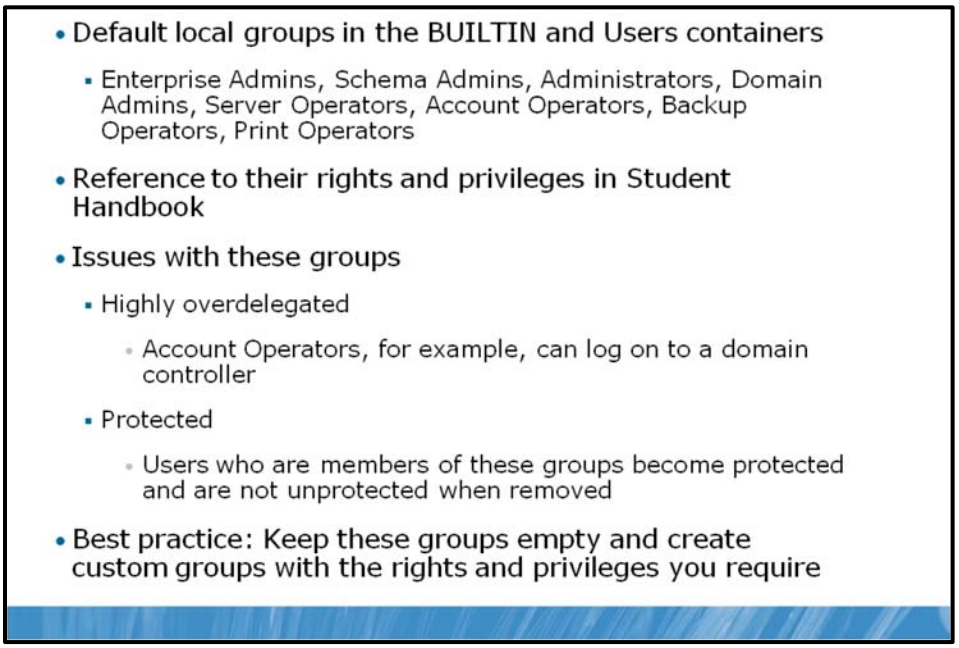
The sales force at Contoso, Ltd. has just completed its fiscal year. Sales files from the previous year are in a folder called, Sales. The sales force needs Read access to the Sales folders. Additionally, a team of auditors from Woodgrove Bank, a potential investor, require Read access to the Sales folders to perform the audit. The following steps are required to implement the security required by this scenario:

1. Assign users with common job responsibilities or other business characteristics to role groups implemented as global security groups. This happens separately in each domain. Sales people at Contoso are added to a Sales role group. Auditors at Woodgrove Bank are added to an Auditors role group.
2. Create a group to manage access to the Sales folders with Read permission. This is implemented in the domain containing the resource that is being managed. In this case, it is the Contoso domain in which the Sales folders reside. The resource access management rule group is created as a domain local group, ACL_Sales Folders_Read.
3. Add the role groups to the resource access management rule group to represent the management rule. These groups can come from any domain in the forest or from a trusted domain such as Woodgrove Bank. Global groups from trusted external domains, or from any domain in the same forest, can be members of a domain local group.
4. Assign the permission that implements the required level of access. In this case, grant the Allow Read permission to the domain local group.

This strategy results in single points of management, reducing the management burden. There is one point of management that defines who is in Sales, or who is an Auditor. Those roles, of course, are likely to have access to a variety of resources beyond simply the Sales folders. There is another single point of

management to determine who has Read access to the Sales folders; and the Sales folders may not just be a single folder on a single server. It could be a collection of folders across multiple servers, each of which assigns the Allow Read permission to the single domain local group.

Default Groups



There are a number of groups that are created automatically on a Windows Server 2008 server. These are called *default local groups*, and they include well-known groups such as Administrators, Backup Operators, and Remote Desktop Users. There are additional groups that are created in a domain, both in the Builtin and Users containers, including Domain Admins, Enterprise Admins, and Schema Admins. The following list provides a summary of capabilities of the subset of default groups that have significant permissions and user rights related to the management of Active Directory.

Enterprise Admins (Users Container of the Forest Root Domain)

This group is a member of the Administrators group in every domain in the forest, giving it complete access to the configuration of all domain controllers. It also owns the Configuration partition of the directory and has full control of the domain naming context in all forest domains.

Schema Admins (Users Container of the Forest Root Domain)

This group owns and has full control of the Active Directory schema.

Administrators (Builtin Container of Each Domain)

This group has complete control over all domain controllers and data in the domain naming context. It can change the membership of all other administrative groups in the domain, and the Administrators group in the forest root domain can change the membership of Enterprise Admins, Schema Admins, and Domain Admins. The Administrators group in the forest root domain is arguably the most powerful service administration group in the forest.

Domain Admins (Users Container of Each Domain)

This group is added to the Administrators group of its domain. It therefore inherits all of the capabilities of the Administrators group. It is also, by default, added to the local Administrators group of each domain member computer, giving Domain Admins ownership of all domain computers.

Server Operators (Built-in Container of Each Domain)

This group can perform maintenance tasks on domain controllers. It has the right to log on locally, start and stop services, perform backup and restore operations, format disks, create or delete shares, and shut down domain controllers. By default, this group has no members.

Account Operators (Built-in Container of Each Domain)

This group can create, modify, and delete accounts for users, groups, and computers located in any OU in the domain (except the Domain Controllers OU), and in the Users and Computers container. Account Operators cannot modify accounts that are members of the Administrators or Domain Admins groups, nor can they modify those groups. Account Operators can also log on locally to domain controllers. By default, this group has no members.

Backup Operators (Built-in Container of Each Domain)

This group can perform backup and restore operations on domain controllers, and log on locally and shut down domain controllers. By default, this group has no members.

Print Operators (Built-in Container of Each Domain)

This group can maintain print queues on domain controllers. It can also log on locally and shut down domain controllers.

The default groups that provide administrative privileges should be managed carefully, because they typically have broader privileges than are necessary for most delegated environments; and because they often apply protection to their members.

The Account Operators group is a perfect example. If you examine its capabilities in the preceding list, you will see that its rights are very broad indeed. It can even log on locally to a domain controller. In very small enterprises, such rights would probably be appropriate for one or two individuals who would probably be domain administrators anyway. In larger enterprises, the rights and permissions granted to Account Operators are usually far too broad.

Additionally, the Account Operators group is, like the other administrative groups, a protected group.

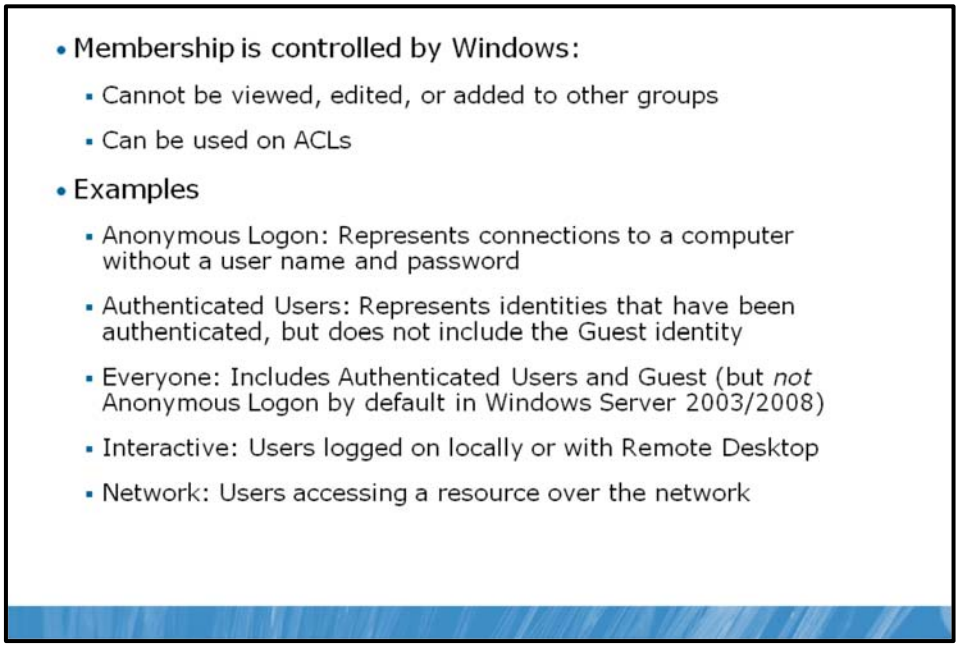
Protected groups are defined by the operating system and cannot be unprotected. Members of a protected group become protected. The result of protection is that the permissions (ACLs) of members are modified so that they no longer inherit permissions from their OU, but rather receive a copy of an ACL that is quite restrictive. For example, if Jeff Ford is added to the Account Operators group, his account becomes protected, and the help desk, which can reset all other user passwords in the Employees OU, cannot reset Jeff Ford's password.

For these reasons of overdelegation and protection, you should strive to avoid adding users to the groups listed above that do not have members by default: Account Operators, Backup Operators, Server Operators, and Print Operators. Instead, create custom groups to which you assign permissions and user rights that achieve your business and administrative requirements.

For example, if Scott Mitchell should be able to perform backup operations on a domain controller, but should not be able to perform restore operations that could lead to database rollback or corruption, and should not be able to shut down a domain controller, do not put Scott in the Backup Operators group.

Instead, create a group and assign it only the Backup Files And Directories user right, then add Scott as a member.

Special Identities

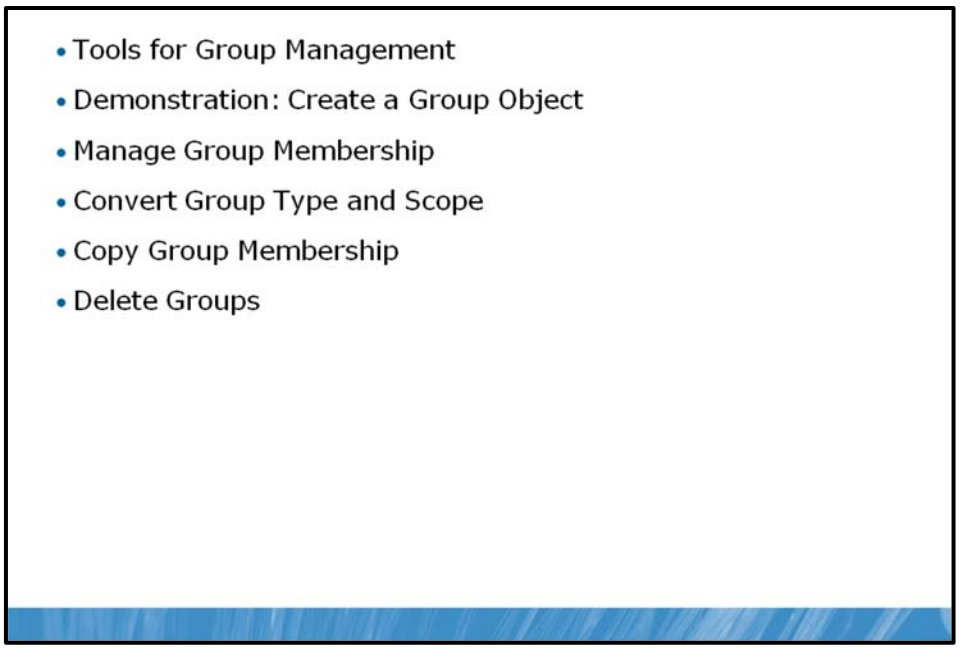


Windows and Active Directory also support special identities, groups for which membership is controlled by the operating system. You cannot view the groups in any list (in the Active Directory Users and Computers snap-in, for example), you cannot view or modify the membership of these special identities, and you cannot add them to other groups. You can, however, use these groups to assign rights and permissions. The most important special identities, often referred to as groups, for convenience, are described in the following list:

- Anonymous Logon. This identity represents connections to a computer and its resources that are made without supplying a user name and password. Prior to Windows Server 2003, this group was a member of the Everyone group. Beginning with Windows Server 2003, this group is no longer a default member of the Everyone group.
- Authenticated Users. This represents identities that have been authenticated. This group does not include Guest, even if the Guest account has a password.
- Everyone. This identity includes Authenticated Users and the Guest account. On computers running versions of Windows earlier than Windows Server 2003, this group includes Anonymous Logon.
- Interactive. This represents users accessing a resource while logged on locally to the computer that is hosting the resource, as opposed to accessing the resource over the network. When a user accesses any given resource on a computer to which the user is logged on locally, the user is automatically added to the Interactive group for that resource. Interactive also includes users logged on through a Remote Desktop connection.
- Network. This represents users accessing a resource over the network, as opposed to users who are logged on locally at the computer that is hosting the resource. When a user accesses any given resource over the network, the user is automatically added to the Network group for that resource.

The importance of these special identities is that they allow you to provide access to resources based on the type of authentication or connection, rather than the user account. For example, you could create a folder on a system that allows users to view its contents when they are logged on locally to the system, but that does not allow the same users to view the contents from a mapped drive over the network. This would be achieved by assigning permissions to the Interactive special identity.

Lesson 2
Administer Groups



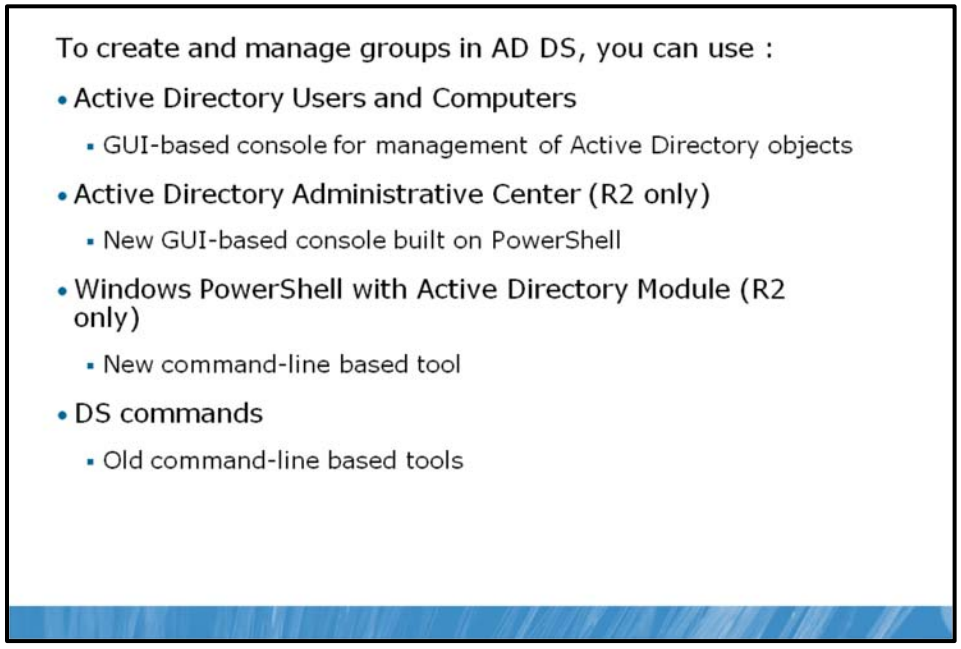
In this lesson, you will learn about the different tools that you can use to manage groups. Using the tools included in Windows Server 2008, you can create and delete group objects, convert group type and scope, and manage group membership.

Objectives

After completing this lesson, you will be able to:

- Create groups with DSADD, CSVDE, and LDIFDE.
- Manage and convert group type and scope.
- Manage group membership with DSMOD and LDIFDE.
- Enumerate group membership with DSGET.
- Delete a group with DSRM.
- Copy group membership.

Tools for Group Management



You can use several GUI-based and command-line tools to create and manage groups in Active Directory Domain Services (AD DS). Each tool provides similar functionality, but the usage scenario will determine which tool is most appropriate. In this topic, we will review the available tools for creating and managing groups.

Active Directory Users and Computers

The Active Directory Users and Computers console is primarily used for group management on a day-to-day basis. It is a GUI-based console and is available in earlier versions of Windows Server. It can be used locally on a domain controller or installed on another server or workstation, and then used remotely. In this console, you can create groups, manage group membership, convert a group from one type to another, and change group scope. Using this console, you can also delete groups, modify group properties, and rename groups. This console is very user-friendly and convenient for simple tasks performed on a relatively small number of group objects.

Note The content in the following sections "Active Directory Administrative Center" and "Windows PowerShell with Active Directory Module" only applies to Windows Server 2008 R2.

Active Directory Administrative Center

In Windows Server 2008 R2, in addition to using Active Directory Users and Computers, administrators can manage their directory service objects by using the new Active Directory Administrative Center.

Built on Windows PowerShell® command-line interface technology, Active Directory Administrative Center provides network administrators with an enhanced Active Directory data management experience and a rich GUI. Administrators can use Active Directory Administrative Center to perform common Active Directory object management tasks through both data-driven navigation and task-oriented navigation.

Although this console provides almost the same functionality as Active Directory Users and Groups when it comes to groups, it is not based on the same technology. In this console, you can use the enhanced GUI to customize Active Directory Administrative Center to meet your particular directory service administering requirements. This can help improve your productivity and efficiency as you perform common Active Directory object management tasks.

Windows PowerShell with Active Directory Module

Windows PowerShell™ is a command-line shell and scripting language that can help information technology (IT) professionals to control system administration more easily and achieve greater productivity.

The Active Directory module for Windows PowerShell in Windows Server 2008 R2 is a Windows PowerShell module named Active Directory that consolidates a group of cmdlets. You can use these cmdlets to manage your Active Directory® domains, Active Directory Lightweight Directory Services (AD LDS) configuration sets, and Active Directory Database Mounting Tool instances in a single, self-contained package.

Using Windows PowerShell, you can manage groups, and perform the following tasks:

- View the permissions of a group by using the **Get-ACL** cmdlet.
- Create a group by using the **New-ADGroup** cmdlet.
- View the nested members of a group by using the **Get-ADGroupMember** cmdlet.
- Move a group within a domain by using the **Move-ADObject** cmdlet.
- Enable Universal group membership caching by using the **Set-ADObject** cmdlet.
- View the direct members of a group by using the **Get-ADGroupMember** cmdlet.
- Modify group attributes by using the **Set-ADGroup** cmdlet.
- Resolve a primary group ID by using the **Get-ADUser** cmdlet.
- Add and remove members of a group by using the **Add-ADGroupMember** or **Remove-ADGroupMember** cmdlets.
- Change the scope or type of a group by using the **Set-ADGroup** cmdlet.
- Restore a deleted group by using the **Restore-ADObject** cmdlet.

For example, if you want to create a global group named, ITAdmins, in the contoso.com domain by using Windows PowerShell, you need to use the following command.

```
New-ADGroup -Name "ITAdmins" -SamAccountName ITAdmins -GroupCategory Security -
GroupScope Global -DisplayName "IT Administrators" -Path "CN=Users,DC=Contoso,DC=Com"
```

If you want to view the direct members of the group, ITAdmins, in the contoso.com domain, you can use following syntax.

```
Get-ADGroupMember ITAdmins | FT Name,ObjectClass -A
```

The following example demonstrates how to move the group SvcAccPSOGroup from the OU Managed to the OU ManagedGroups in the contoso.com domain.

```
Move-ADObject "CN= SvcAccPSOGroup,OU=Managed,DC=Contoso,DC=Com" -TargetPath
"OU=ManagedGroups,DC=Contoso,DC=Com"
```

The following example demonstrates how to add the user, SaraDavis, to the group, SvcAccPSOGroup.

```
Add-ADGroupMember -Identity SvcAccPSOGroup -Member SaraDavis
```

Note For a full explanation of the parameters that you can pass to any cmdlet in Windows PowerShell, at the Active Directory module command prompt, type Get-Help cmdletname –detailed and then press Enter.

DS commands

In previous versions of Windows Server, such as Windows Server 2003, where PowerShell was not included, other type of command-line utilities were used to manage Active Directory objects.

These command-line tools were provided with server operating systems to allow better and more productive management of the directory service. These tools are called DS commands.

The following is a list of DS commands and their functionality:

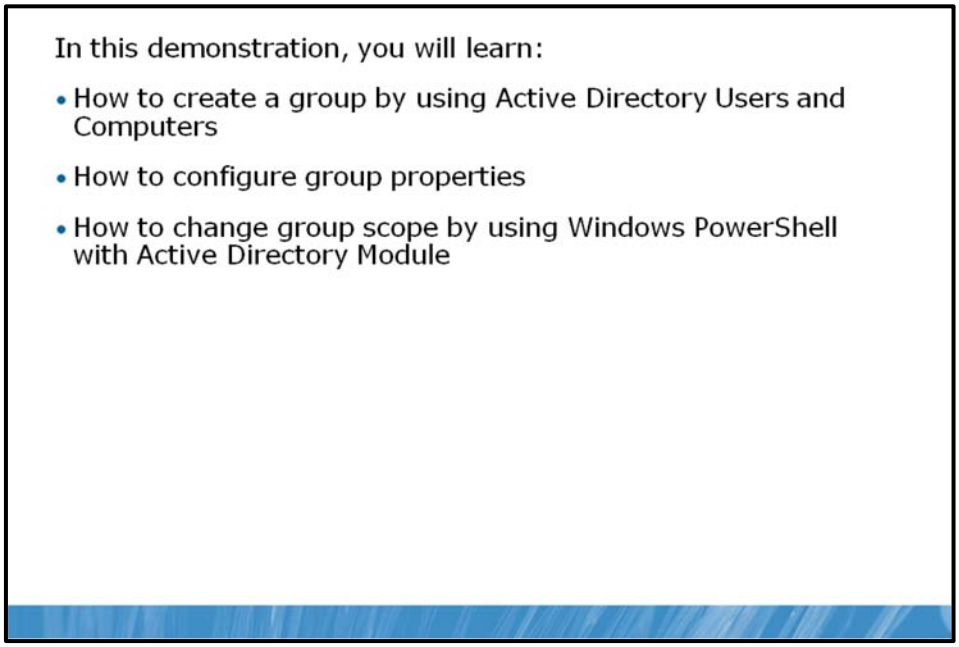
- **DSGet**. Returns the current value of the specified directory object property
- **DSQuery.** Allows the directory service to be searched for an object or all objects with like properties
- **DSMod.** Helps an administrator change properties for existing directory objects
- **DSrm.** Removes objects from the directory
- **DSAdd.** Allows administrators to add new directory objects
- **DSMove.** Allows objects to be moved from one OU to another

These commands can be also used in Windows Server 2008 R2 to manage groups. However, because Windows Server 2008 R2 includes a newer and more powerful command-line based environment, these tools are used to support legacy scripts.

For example, to create a new global security group named, Marketing, the following command would be used.

```
dsadd group "CN=Marketing,OU=Role,OU=Groups,DC=contoso,DC=com"
    –samid Marketing –secgrp yes –scope g
```

Demonstration: Create a Group Object

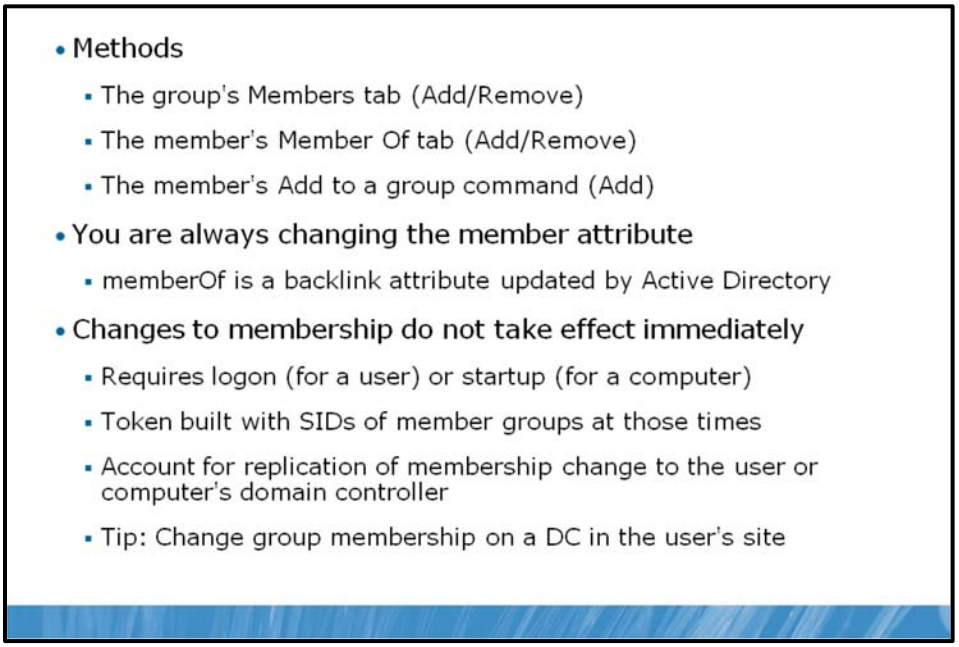


Groups are an important class of object, because they are used to collect users, computers, and other groups to create a single point of management. The most straightforward and common use of a group is to grant permissions to a shared folder. For example, if a group has been given the Read access to a folder, any of the group's members will be able to read the folder. You do not have to grant Read access directly to each individual member—you can manage access to the folder simply by adding and removing members of the group.

Demonstration steps:

- Create a group by using Active Directory Users and Computers.
- Configure group properties.
- Change group scope by using Windows PowerShell with Active Directory Module.

Manage Group Membership

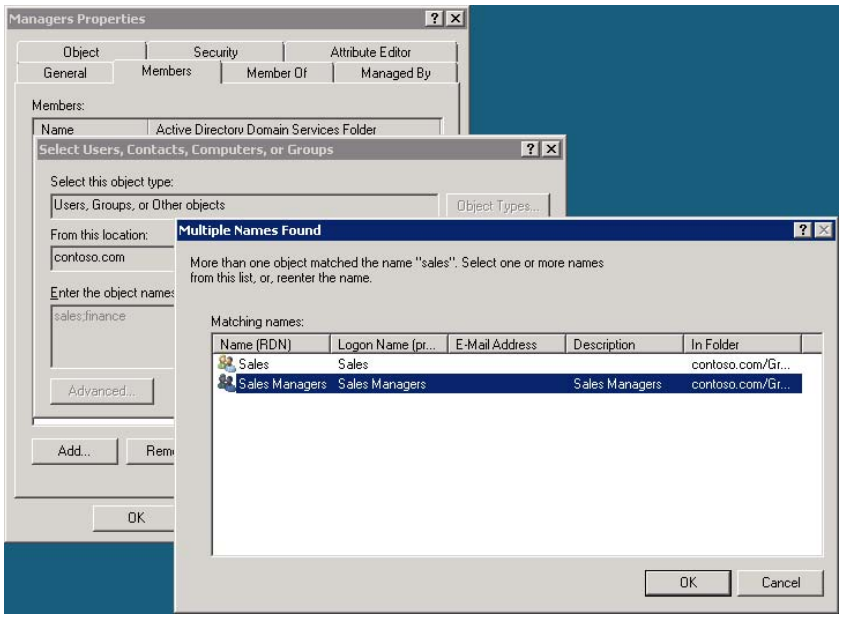


You can add or remove members of a group by using several methods. These include using the **Members** tab, the **Member of** tab, the **Add to a group** command, and the **Member** and **MemberOf** Attributes.

The Members Tab

To manage group membership by using the group's **Members** tab:

1. Open the group's **Properties** dialog box.
2. Click the **Members** tab.
3. To remove a member, simply select the member and click **Remove**.
4. To add a member, click the **Add** button. The **Select Users, Computers, Service Accounts, or Groups** dialog box appears, as follows:



There are several tips worth mentioning about this process:

- In the **Select** dialog box, in the **Enter The Object Names** box, you can type multiple accounts separated by semicolons. For example, in the screenshot shown above, both sales and finance were entered. They are separated by a semicolon.
- You can type partial names of accounts—you do not need to type the full name. Windows searches Active Directory for accounts that begin with the name you entered. If there is only one match, Windows selects it automatically. If there are multiple accounts that match, the **Multiple Names Found** dialog box appears, allowing you to select the specific object you want. This shortcut—typing partial names—can save time when you are adding members to groups and can help when you don't remember the exact name of a member.
- By default, Windows searches only for users and groups that match the names you enter in the Select dialog box. If you want to add computers to a group, you must click the **Options** button and select **Computers**.
- By default, Windows searches only domain groups. If you want to add local accounts, click the **Locations** button on the **Select** dialog box.
- If you cannot find the member you want to add, click the **Advanced** button on the **Select** dialog box. A more powerful query window will appear, giving you more options for searching Active Directory.

The Member Of Tab

To manage group membership by using the member object's **Member Of** tab:

1. Open the properties of the member object, and then click its **Member Of** tab.
2. To remove the object from a group, select the group and then click the **Remove** button.
3. To add the object to a group, click the **Add** button, and then select the group.

The Add to a group Command

To manage group membership by using the **Add to a group** command:

1. Right-click one or more selected objects in the Active Directory Users and Computers details pane.
2. Click the **Add to a group** command.
3. Use the **Select** dialog box to specify the group.

The Member and MemberOf Attributes

When you add a member to a group, you change the group's member attribute. The member attribute is a multivalued attribute. Each member is a value represented by the distinguished name of the member. If the member is moved or renamed, Active Directory automatically updates the member attributes of groups that include the member.

When you add a member to a group, the member's **memberOf** attribute is also updated, indirectly. The **memberOf** attribute is a special type of attribute called a backlink. It is updated by Active Directory when a forward link attribute, such as member, refers to the object. When you add a member to a group, you are always changing the member attribute. Therefore, when you use the **Member Of** tab of an object to add to a group, you are actually changing the group's member attribute. Active Directory updates the **memberOf** attribute automatically.

Helping Membership Changes Take Effect Quickly

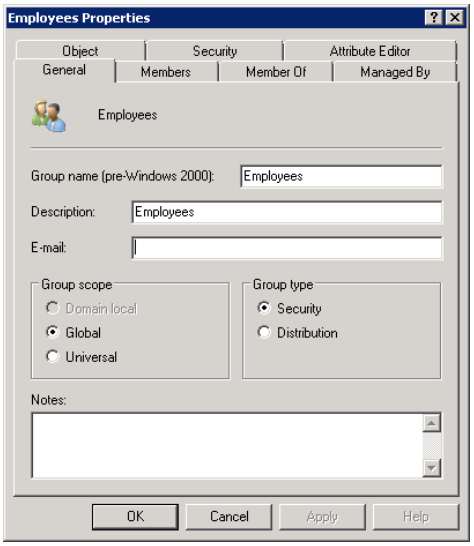
When you add a user to a group, the membership does not take effect immediately. Group membership is evaluated at logon for a user (at startup for a computer). Therefore, a user will have to log off and log on before the membership change becomes a part of the user's token.

Additionally, there may be a delay while the group membership change replicates. (Replication will be discussed in Module 12.) This is particularly true if your enterprise has more than one Active Directory site. You can facilitate the speed with which a change impacts a user by making the change on a domain controller in the user's site. Right-click the domain in the Active Directory Users and Computers snap-in, and then click **Change Domain Controller**.

Convert Group Type and Scope

- In Active Directory Users and Computers, you can change group type:
  - Security to distribution (* lose permissions assigned to group)
  - Distribution to security
- In Active Directory Users and Computers, you can change the group scope:
  - Global to universal
  - Domain local to universal
  - Universal to global
  - Universal to domain local
  - You cannot change DL → G or G → DL directly, but you *can* change DL → U → G *or* G → U → DL.
  - Change prevented if memberships are invalid—fix, then retry
- `dsmod group GroupDN -secgrp { yes | no } -scope { l | g | u }`

If, after creating a group, you determine that you need to modify the group's scope or type, you can do so. Open the **Properties** of an existing group, and on the **General** tab, shown in the following image, you will see the existing scope and type. At least one more scope and type are available to be selected.



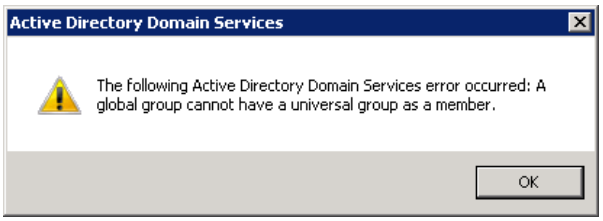
You can convert the group type at any time by changing the selection in the **Group Type** section of the **General** tab. Be cautious, however; when you convert a group from security to distribution, any resources to which the group had been assigned permission will no longer be accessible in the same way. After the group becomes a distribution group, users who log on to the domain will no longer include the group's SID in their security access tokens.

You can change the group scope in one of the following ways:

- Global to Universal
- Domain local to Universal
- Universal to Global
- Universal to Domain local

The only scope changes that you cannot make directly are from global to domain local or domain local to global. However, you can make these changes indirectly by first converting to universal scope, then converting to the desired scope. So, all scope changes are possible.

Remember, however, that a group's scope determines the types of objects that can be members of the group. If a group already contains members, or is a member of another group, you will be prevented from changing the scope. For example, if a global group is a member of another global group, you cannot change the first group to universal scope, because a universal group cannot be a member of a global group. You will be given an explanatory error message, such as that shown below. You must correct the membership conflicts before you can change the group's scope.



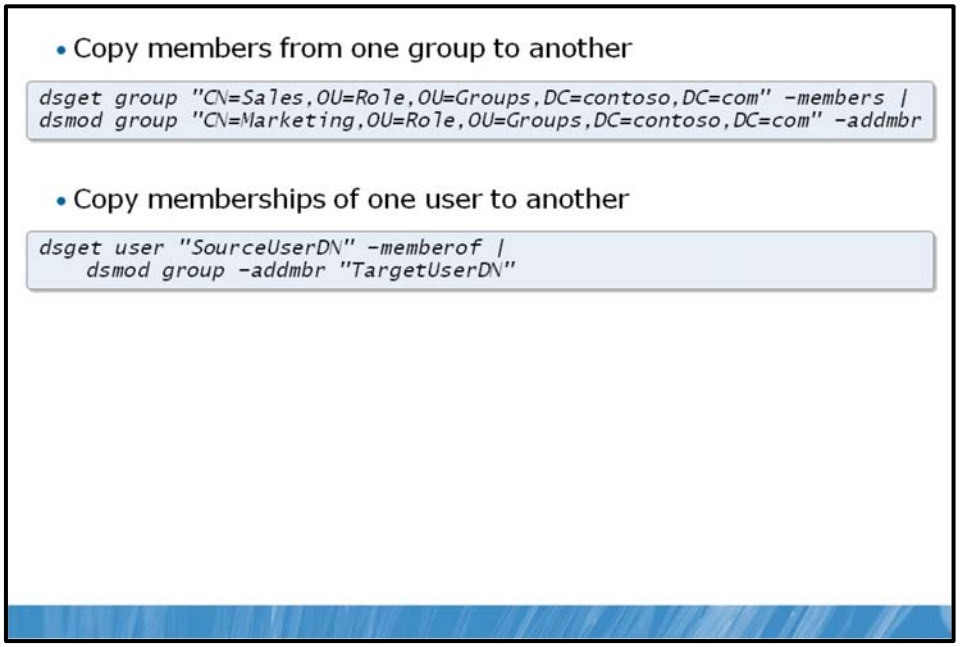
The DSMod command can be used to change group type and scope by using the following syntax.

```
dsmod group GroupDN –secgrp { yes | no } –scope { l | g | u }
```

The *GroupDN* is the distinguished name of the group to modify. The following two parameters affect group scope and type.

- **-secgrp { yes | no }.** Specifies group type: security (*yes*) or distribution (*no*)
- **-scope { l | g | u }.** Determines the group scope: domain local (*l*), global (*g*), or universal (*u*)

Copy Group Membership



You can use **DSGet** in combination with DSMod to copy group membership. In the following example, the **DSGet** command is used to get information about all the members of the Sales group, and then, by piping that list to DSMod, to add those users to the Marketing group.

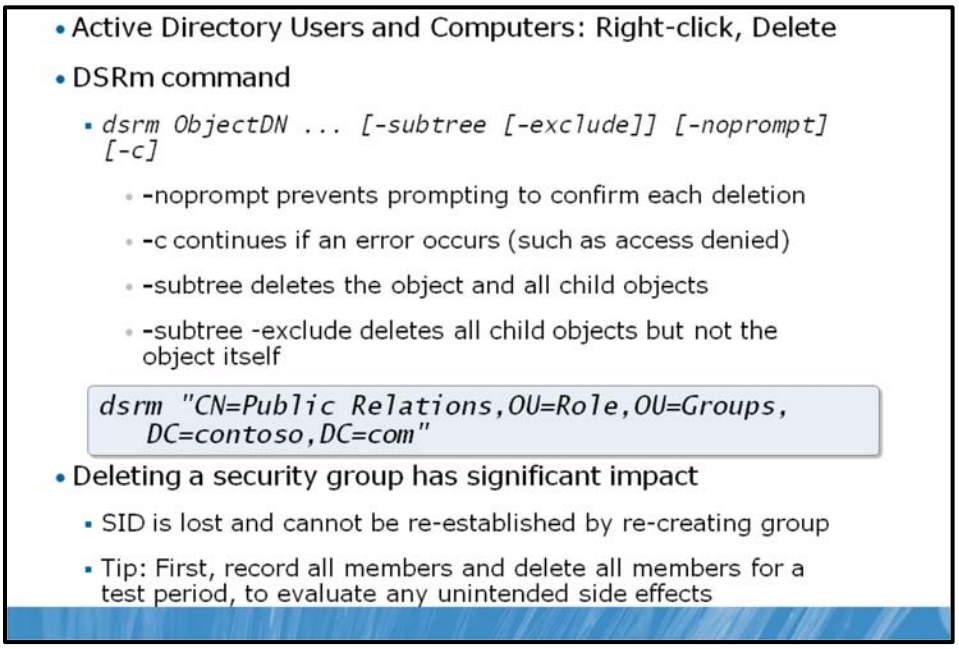
```
dsget group "CN=Sales,OU=Role,OU=Groups,DC=contoso,DC=com" –members |
dsmod group "CN=Marketing,OU=Role,OU=Groups,DC=contoso,DC=com" –addmbr
```

Notice the use of piping. The "output" of **DSGet** (distinguished names of members of the first group) is piped, using the pipe symbol ("|"), to act as the "input" for the DNs that are missing from the **-**addmbr switch.

Similarly, the **DSGet** and DSMod commands can work together to copy the group membership of one object, such as a user, to another object.

```
dsget user "SourceUserDN" –memberof |
     dsmod group –addmbr "TargetUserDN"
```

Delete Groups



You can delete a group in the Active Directory Users and Computers snap-in by right-clicking the group and choosing the **Delete** command.

Also, **DSrm** can be used to delete a group or any other Active Directory object. The basic syntax of DSRm is as follows.

```
dsrm ObjectDN ... [-subtree [-exclude]] [-noprompt] [-c]
```

The object is specified by its distinguished name in the *ObjectDN* parameter. You will be prompted to confirm the deletion of each object, unless you specify the *-noprompt* option. The *-c* switch puts DSRm into continuous operation mode, in which errors are reported but the command keeps processing additional objects; without the *-c* switch, processing halts on the first error.

The *-subtree* option causes DSrm to delete the object and all child objects. The *-subtree -exclude* option will delete all child objects, but not the object itself.

To delete the Public Relations group, type the following command.

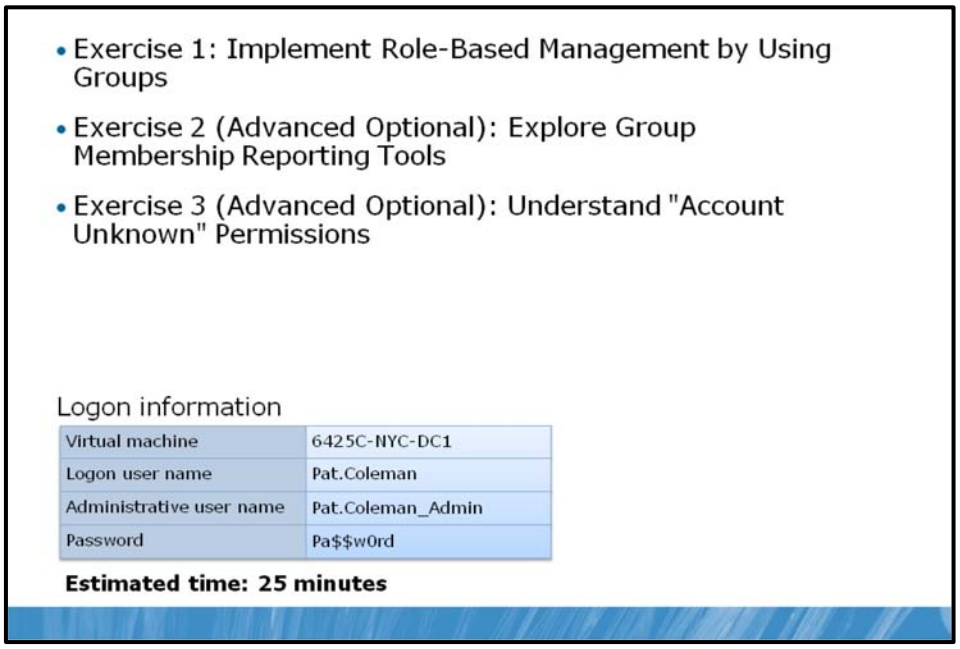
```
dsrm "CN=Public Relations,OU=Role,OU=Groups,DC=contoso,DC=com"
```

Know the Impact Before Deleting a Group

When you delete a group, you are removing a point of management in your organization. Be certain you have evaluated the environment to know that there are no permissions or other resources that rely on the group. Deleting a group is a serious action with potentially significant consequences. When you delete a group, you remove its SID. Re-creating the group with the same name does not restore permissions, because the new group's SID is different than that of the original group.

We recommend that before you delete a group, you record its membership and remove all members for a period of time, to determine whether the members lose access to any resources. If anything goes wrong, simply re-add the members. If the test succeeds, then delete the group.

Lab A: Administer Groups



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In Hyper-V™ Manager, click **6425C-NYC-DC1**, and in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Log on by using the following credentials:
   - User name: **Pat.Coleman**
   - Password: **Pa$$w0rd**
5. Open Windows Explorer and then browse to **D:\Labfiles\Lab04a**.
6. Run **Lab04a_Setup.bat** with administrative credentials. Use the account **Pat.Coleman_Admin** with the password **Pa$$w0rd**.
7. The lab setup script runs. When it is complete, press any key to continue.
8. Close the Windows Explorer window, **Lab04a**.

Lab Scenario

To improve the manageability of resource access at Contoso, Ltd., you have decided to implement role-based management. The first application of role-based management will be to manage who can access the folders containing sales information. You must create groups that manage access to that sensitive information. Business rules are that Sales and Marketing employees, and a team of Consultants, should be able to read the Sales folders. Additionally, Bobby Moore requires Read access. Finally, you have been

asked to discover a way to produce a list of group members, including those who are in nested groups; and a list of a user's group membership, including indirect or nested membership.

Exercise 1: Implement Role-Based Management by Using Groups

In this exercise, you will implement role-based management by using groups and the best practice group nesting strategy, IGDLA. You will create different scopes and types by using both the Active Directory Users and Computers snap-in, and command-line tools.

The main tasks for this exercise are as follows:

1. Create role groups with Active Directory Users and Computers.
2. Create role groups with DSAdd.
3. Add users to the role group.
4. Implement a role hierarchy in which Sales Managers are also part of the Sales role.
5. Create a resource access management group.
6. Assign permissions to the resource access management group.
7. Define which roles and users have access to a resource.

▶ Task 1: Create role groups with Active Directory Users and Computers.

1. Run **Active Directory Users and Computers** with administrative credentials. Use the account **Pat.Coleman_Admin**, with the password, **Pa$$w0rd**.
2. Create global security groups called, **Sales** and **Consultants**, in the **Groups\Role** OU.

▶ Task 2: Create role groups with DSAdd.

1. Run **command prompt** with administrative credentials. Use the account, **Pat.Coleman_Admin**, with the password, **Pa$$w0rd**.
2. Using the **DSAdd** command, create a global security group named, **Auditors**, in the **Groups\Role** OU.
3. In **Active Directory Users and Computers**, confirm that the object has been created.

▶ Task 3: Add users to the role group.

1. Add **Tony Krijnen** to the **Sales** group by using the **Members** tab of the **Sales** group.
2. Add **Linda Mitchell** to the **Sales** group by right-clicking **Linda Mitchell** and choosing **Add to a group**.

▶ Task 4: Implement a role hierarchy in which Sales Managers are also part of the Sales role.

- Add the **Sales Managers** group as a member of the **Sales** group by using the **Member Of** tab of the **Sales Managers** group.

▶ Task 5: Create a resource access management group.

- Create a domain local security group named, **ACL_Sales Folders_Read**, in the **Groups\Access** OU.

▶ Task 6: Assign permissions to the resource access management group.

1. Verify that there is a folder in D:\Data named, **Sales**.
2. Right-click the **Sales** folder, click **Properties**, and then click the **Security** tab.

3. Click **Edit**, and then click **Add**.
4. Type **ACL_** and press ENTER.

   Notice that when you use a prefix for group names, such as the ACL_ prefix for resource access groups, you can find them quickly.
5. Click **ACL_Sales Folders_Read**, and then click **OK**.
6. Confirm that the group has been given Read & execute permission.
7. Click **OK** to close each open dialog box.

▶ Task 7: Define the roles and users that have access to a resource.

- Add **Sales**, **Consultants**, **Auditors**, and **Bobby Moore** to the **ACL_Sales Folders_Read** group.

Results: In this exercise, you implemented simple role-based management to manage Read access to the Sales folder.

Exercise 2 (Advanced Optional): Explore Group Membership Reporting Tools

Advanced Optional exercises provide additional challenges for students who are able to complete lab exercises quickly. There are no answers in the Lab Answer Key.

The main tasks for this exercise are as follows:

1. Open **D:\AdminTools\Members_Report.hta**. Enter the name of a group, and then click **SHOW MEMBERS**.
2. Open **D:\AdminTools\MemberOf_Report.hta**. Enter the name of a user, computer, or group, and then click **Report**.

Exercise 3 (Advanced Optional): Understand "Account Unknown" Permissions

Advanced Optional exercises provide additional challenges for students who are able to complete lab exercises quickly. There are no answers in the Lab Answer Key.

The main tasks for this exercise are as follows:

1. In the **Role** OU, create a global security group named, **Test**.
2. Give the group **Read & Execute** permission to the **D:\Data\Sales** folder.
3. Delete the group named, **Test**.
4. Examine the **Security** tab of the Sales folder's properties dialog box. If you still see the Test group listed, Windows Explorer may be caching the mapping of the SID to the group name. Log off, log on, and check again.

Note Do not shut down the virtual machines after you finish this lab because the settings you have configured here will be used in Lab B.

Lab Review Questions

Question: Describe the purpose of global groups in terms of role-based management.

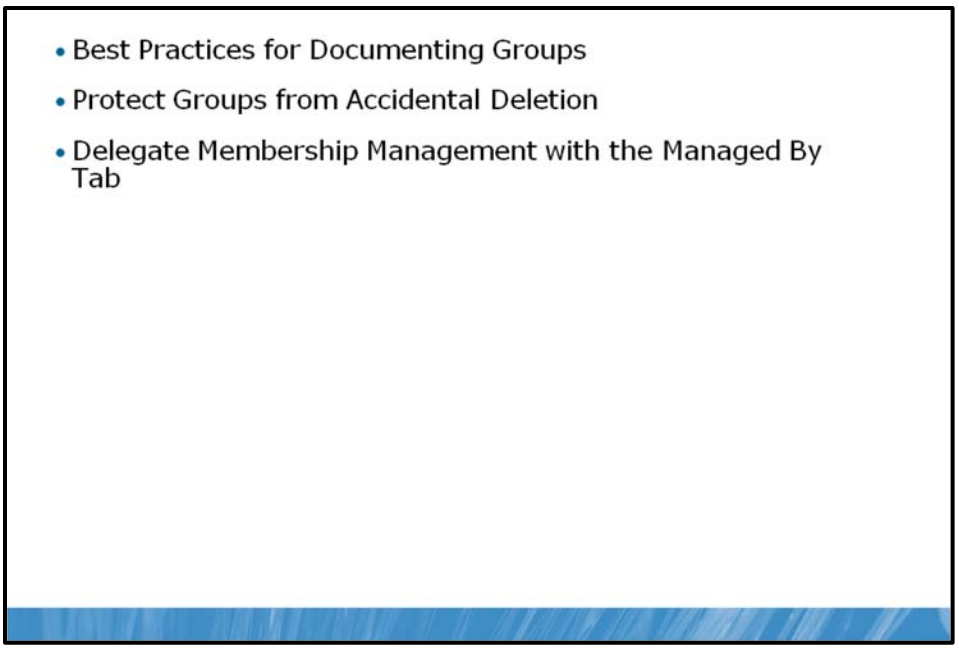
Question: What types of objects can be members of global groups?

Question: Describe the purpose of domain local groups in terms of role-based management of resource access.

Question: What types of objects can be members of domain local groups?

Question: If you have implemented role-based management and are asked to report who can read the Sales folders, what command would you use to do so?

Lesson 3
Best Practices for Group Management



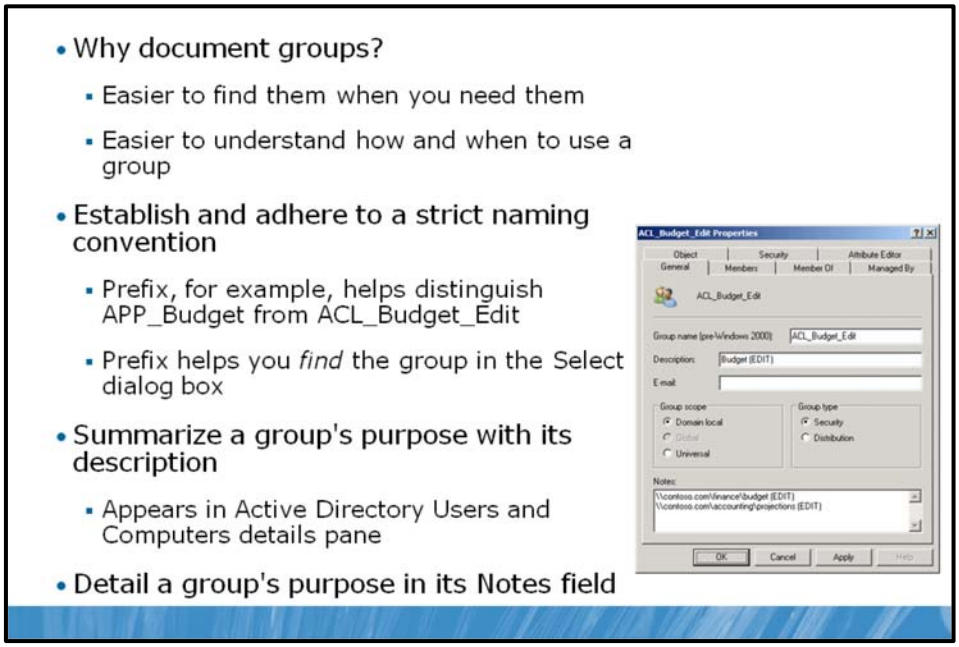
In this lesson, you will learn about the best practices that you should follow when you manage groups. You will also see how to protect groups from accidental deletion and how to use the Managed By tab to delegate membership management.

Objectives

After completing this lesson, you will be able to:

- Describe the best practices for group documentation.
- Protect a group from accidental deletion.
- Delegate group membership management by using the Managed By tab.

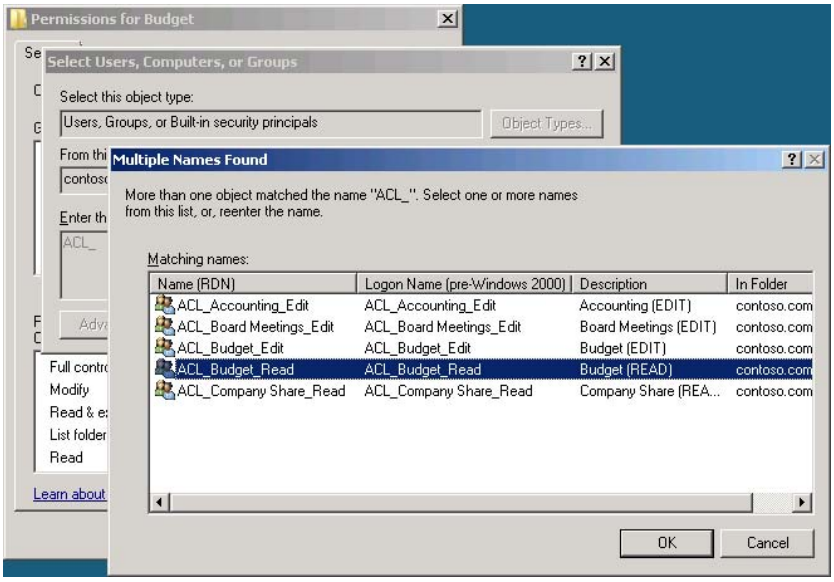
Best Practices for Documenting Groups



Creating a group in Active Directory is easy. It is not so easy to ensure that the group is used correctly over time. You can facilitate the correct management and use of a group by documenting its purpose, to help administrators understand how and when to use the group. There are several best practices that will prove immensely useful to your enterprise group administration.

Establish and Adhere to a Strict Naming Convention

An earlier lesson dealt with a suggested naming convention. In the context of ongoing group administration, establishing and following group naming standards increases administrative productivity. Using prefixes to indicate the purpose of a group, and using a consistent delimiter between the prefix and the descriptive part of the group name can help users locate the correct group for a particular purpose. For example, the prefix APP can be used to designate groups that are used to manage applications, and the prefix ACL can be used for groups that are assigned permissions on access control lists (ACLs). With such prefixes, it becomes easier to locate and interpret the purpose of groups named, for example, APP_Accounting versus ACL_Accounting_Read—the former is used to manage the deployment of the accounting software, and the latter to provide Read access to the accounting folder. Prefixes also help to group the names of groups in the user interface as illustrated in the example shown in the following screen shot.



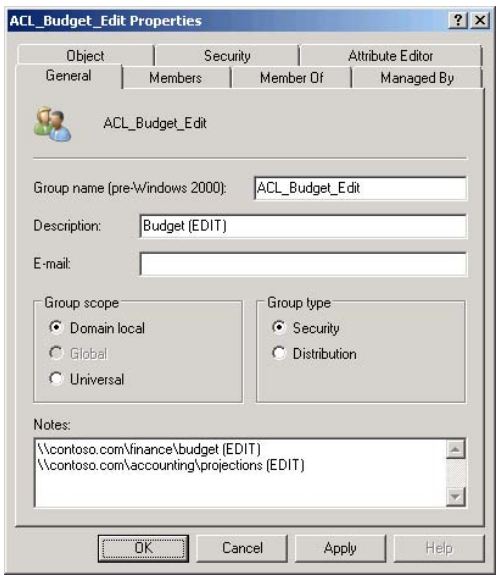
When attempting to locate a group to use in assigning permissions to a folder, you can type the prefix, ACL,_in the **Select** dialog box and click **OK**. A **Multiple Items Found** dialog box appears showing only the ACL_ groups in the directory, thereby ensuring that permissions will be assigned to a group that is designed to manage resource access.

Summarize a Group's Purpose with its Description Attribute

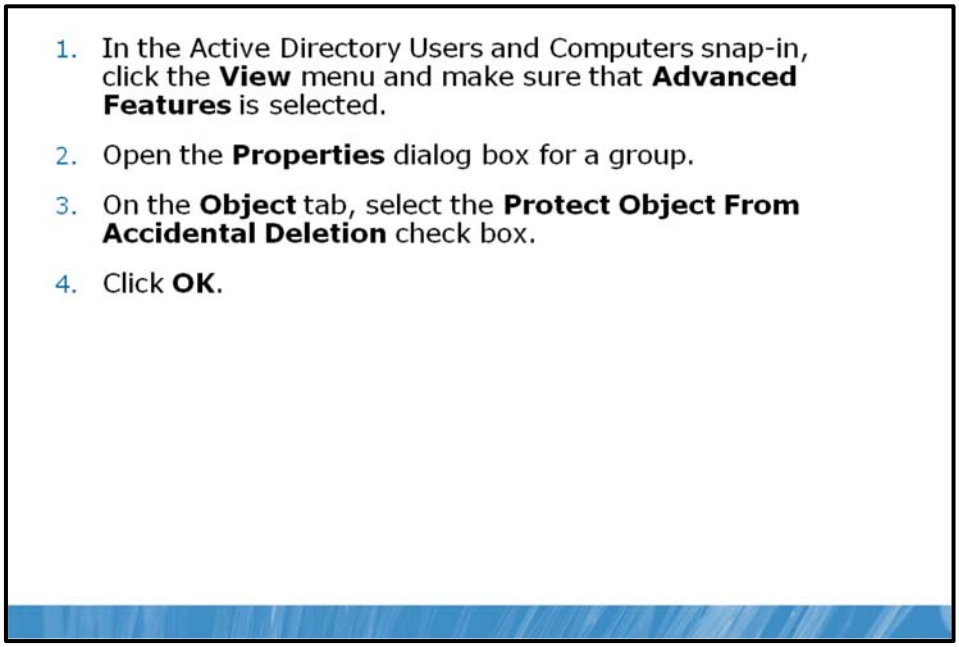
Use the **Description** attribute of a group to summarize the group's purpose. Because the **Description** column is enabled by default in the details pane of the Active Directory Users and Computers snap-in, the group's purpose can be highly visible to administrators.

Detail a Group's Purpose in its Notes

When you open a group's **Properties** dialog box, the **Notes** field is visible at the bottom of the **General** tab. This field can be used to record the group's purpose. For example, you can list the folders to which a group has been given permission, as follows.



Protect Groups from Accidental Deletion



Protect yourself from the potentially devastating results of deleting a group by protecting each group you create from deletion. Windows Server 2008 makes it easy to protect any object from accidental deletion.

To protect an object, perform the following steps:

1. In the **Active Directory Users and Computers** snap-in, click the **View** menu and ensure that **Advanced Features** is selected.
2. Open the **Properties** dialog box for a group.
3. On the **Object** tab, select the **Protect Object From Accidental Deletion** check box.
4. Click **OK**.

This is one of the few places in Windows in which you actually have to click **OK**. Clicking **Apply** does not modify the ACL based on your selection.

The **Protect Object From Accidental Deletion** option applies an access control entry (ACE) to the ACL of the object that explicitly denies the Everyone group both the Delete permission and the Delete Subtree permission. If you really do want to delete the group, you can return to the **Object** tab of the **Properties** dialog box and clear the **Protect Object From Accidental Deletion** check box.

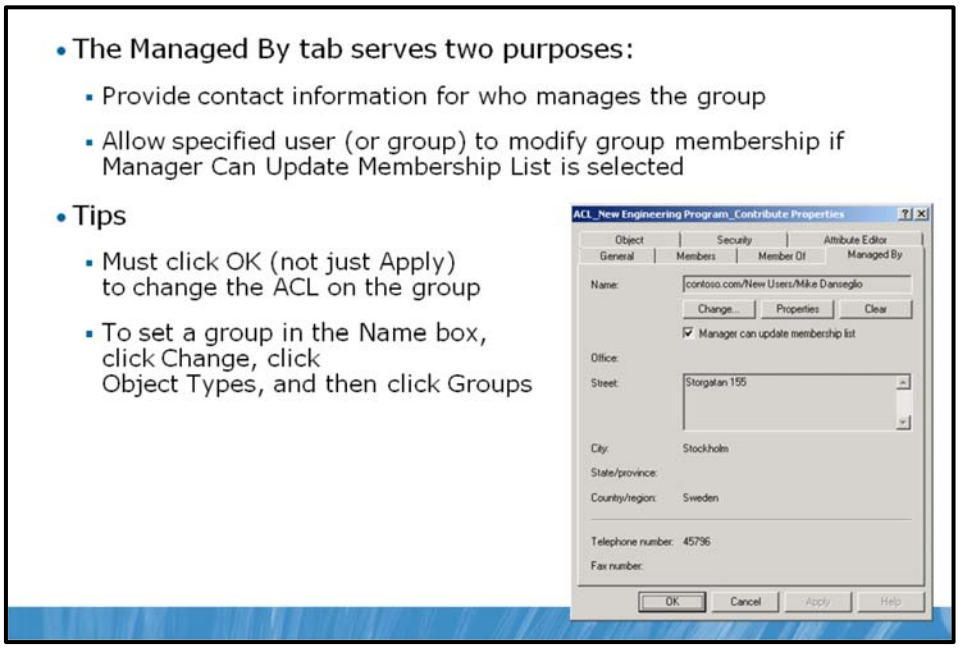
Deleting a group has a high impact on administrators, and potentially, on security. Consider a group that has been used to manage access to resources. If the group is deleted, access to that resource is changed. Either users who should be able to access the resource are suddenly prevented from access, creating a denial-of-service scenario, or if you had used the group to deny access to a resource with a Deny permission, inappropriate access to the resource becomes possible.

Additionally, if you re-create the group, the new group object will have a new SID, which will not match the SIDs on ACLs of resources. So you must instead perform object recovery to reanimate the deleted group before the tombstone interval is reached. When a group has been deleted for the tombstone interval—60 days, by default—the group and its SID are permanently deleted from Active Directory.

When you reanimate a tombstoned object, you must re-create most of its attributes, including importantly, the member attribute of group objects. That means, you must rebuild the group membership after restoring the deleted object. Alternatively, you can perform an authoritative restore; or, in Windows Server 2008, turn to your Active Directory snapshots to recover both the group and its membership. Authoritative restore and snapshots are discussed in Module 13.

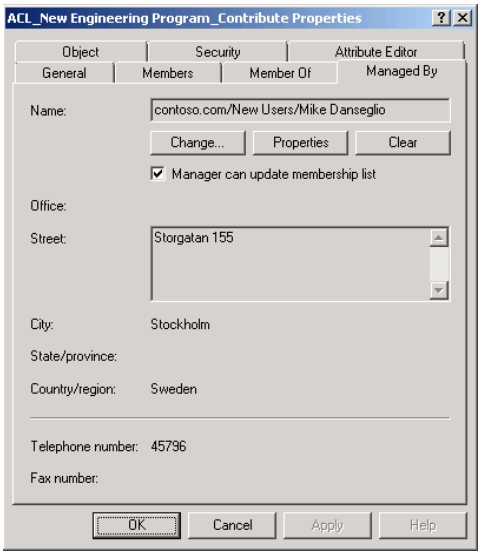
In any event, it is safe to say that recovering a deleted group is a skill you should hope to use only in disaster recovery fire drills, not in a production environment.

Delegate Membership Management with the Managed By Tab



After a group has been created, you might want to delegate the management of the group's membership to a team or an individual who has the business responsibility for the resource that the group manages. For example, let's assume that your finance manager is responsible for creating next year's budget. You create a shared folder for the budget and assign Write permission to a group named, ACL_Budget_Edit. If someone needs access to the budget folder, he or she contacts the help desk to enter a request, the help desk contacts the finance manager for business approval, and then the help desk adds the user to the ACL_Budget_Edit group. You can improve the responsiveness and accountability of the process by allowing the finance manager to change the group's membership. Then, users who need access can request access directly from the finance manager, who can make the change, thus removing the intermediate step of contacting the help desk. To delegate the management of a group's membership, you must assign to the finance manager the Allow Write Member permission for the group. The member attribute is the multivalued attribute that is the group's membership.

The easiest way to delegate membership management of a single group is to use the **Managed By** tab. The **Managed By** tab of a group object's **Properties** dialog box is shown here:

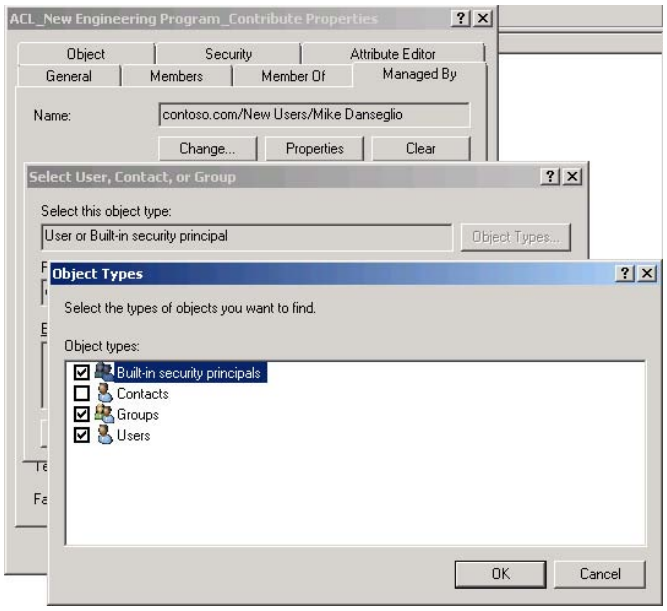


The **Managed By** tab serves two purposes. First, it provides contact information related to the manager of a group. You can use this information to contact the business owner of a group to obtain approval before adding a user to the group.

The second purpose served by the **Managed By** tab is to manage the delegation of the member attribute. Note the check box shown in the preceding screenshot. It is labeled **Manager can update membership list**. When selected, the user or group shown in the **Name** box is given the **Allow Write Member** permission. If you change or clear the manager, the appropriate change is made to the group's ACL.

Tip You must actually click OK to implement the change. Clicking Apply does not change the ACL on the group.

It is not quite so easy to insert a group into the **Managed By** tab of another group. When you click the **Change** button, the **Select User, Contact, Or Group** dialog box appears. If you enter the name of a group and click **OK**, an error occurs. That is because this dialog box is not configured to accept groups as valid object types, even though "Group" is in the name of the dialog box itself. To work around this odd limitation, click the **Object Types** button, and then select the check box next to Groups. Click **OK** to close both the **Object Types** and **Select** dialog boxes. Ensure to select the **Manager Can Update Membership List** check box if you want to assign the Allow Write Member permission to the group. When a group is used on the **Managed By** tab, no contact information is visible, because groups do not maintain contact-related attributes.

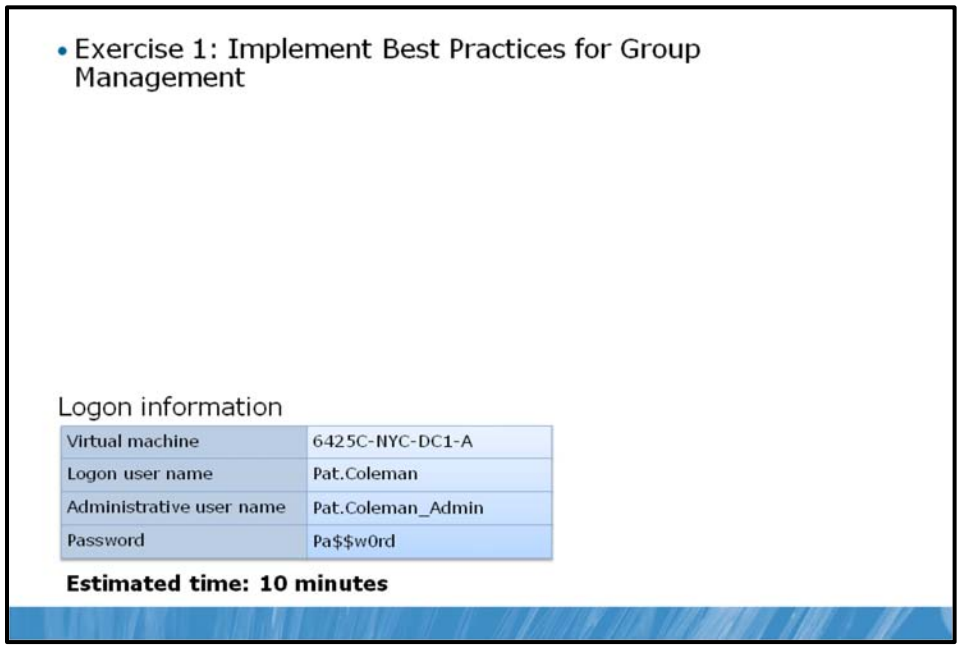


After you have delegated group membership management, a user does not require Active Directory Users and Computers to modify the membership of the group. A user can simply use the **Search Active Directory** capability of Windows clients to find the group, and then change its membership.

To find a group:

1. Click **Start**, and then click **Network**.
2. Click the **Search Active Directory** button on the toolbar.
3. Type the name of the group and click **Find Now**.

Lab B: Best Practices for Group Management



Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In Hyper-V™ Manager, click **6425C-NYC-DC1**, and in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Log on by using the following credentials:
   - User name: **Pat.Coleman**
   - Password: **Pa$$w0rd**

Lab Scenario

Your implementation of role-based management at Contoso, Ltd. has been highly successful. As the number of groups in the domain has increased, you've come to realize that it is important to record the groups and prevent administrators from accidentally deleting a group. Finally, you want to allow the business owners of resources to manage access to those resources by delegating to those owners the right to modify the membership of appropriate groups.

Exercise 1: Implement Best Practices for Group Management

In this exercise, you will perform the following tasks to record, delegate, and secure groups:

1. Create a well-documented group.
2. Protect a group from accidental deletion.
3. Delegate group membership management.
4. Validate the delegation of group membership management.

▶ Task 1: Create a well-documented group.

1. Run **Active Directory Users and Computers** with administrative credentials. Use the account, **Pat.Coleman_Admin**, with the password, **Pa$$w0rd**.
2. Browse to the **Groups\Access** OU. In the properties of the **ACL_Sales Folders_Read** group, configure the following:
   - A **Description** that summarizes the resource management rule represented by the group: **Sales Folders (READ)**
   - In the **Notes** box, type the following paths to represent the folders that have permissions assigned to this group.
     \\contoso\teams\Sales (READ)
     \\file02\data\Sales (READ)
     \\file03\news\Sales (READ)

▶ Task 2: Protect a group from accidental deletion.

1. Enable the **Advanced Features** view of the Active Directory Users and Computers snap-in.
2. Protect the **ACL_Sales Folders_Read** group from being accidentally deleted.
3. Attempt to delete the group. Confirm that the attempt to delete the group is denied.

▶ Task 3: Delegate group membership management.

- Configure the **Managed By** attribute of **Auditors** to refer to **Mike Danseglio**.

▶ Task 4: Validate the delegation of group membership management.

1. Log off from NYC-DC1, then log on with user name, **Mike.Danseglio**, and the password, **Pa$$w0rd**.
2. Open the **Network** window and use **Search Active Directory** to locate the **Auditors** group.
3. Add the **Executives** group to the **Auditors** group.
4. Log off from NYC-DC1.

Results: In this exercise, you created a well-documented group, protected it from accidental deletion, and delegated group membership management.

▶ To prepare for the next module

When you finish the lab, revert the virtual machines to their initial state. To do this, complete the following steps:

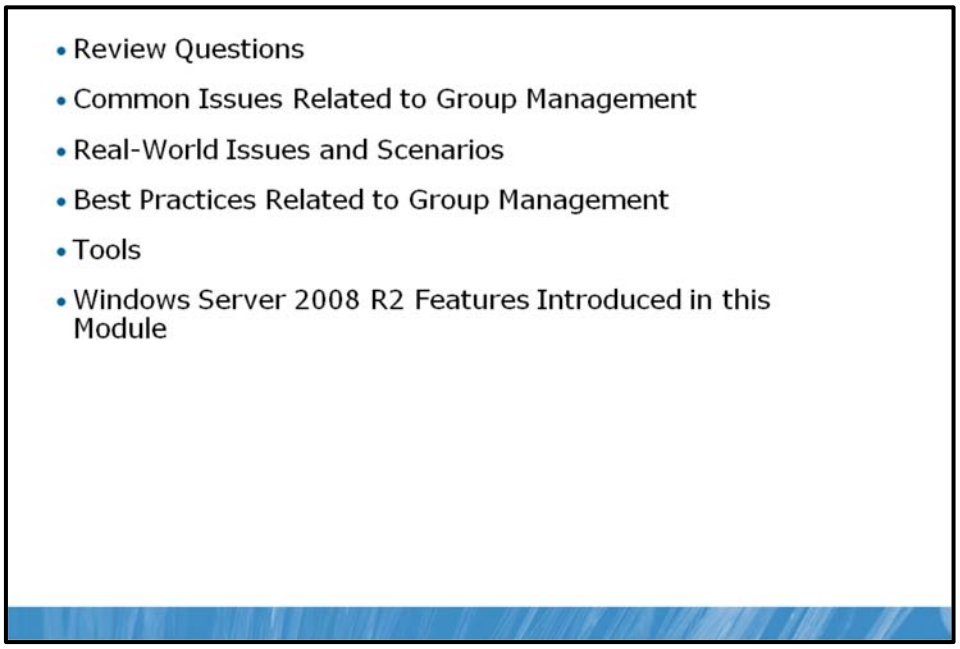
1. On the host computer, start Hyper-V Manager.
2. Right-click **6425B-NYC-DC1** in the **Virtual Machines** list, and then click **Revert**.
3. In the **Revert Virtual Machine** dialog box, click **Revert**.

Lab Review Questions

Question: What are some benefits of using the Description and Notes fields of a group?

Question: What are the advantages and disadvantages of delegating group membership?

Module Review and Takeaways



Review Questions

1. Members of a Sales department in a company that has branches in multiple cities travel frequently between domains. How will you provide these members with access to printers on various domains that are managed by using domain local groups?
2. You are responsible for managing accounts and access to resources for your group members. A user in your group transfers into another department within the company. What should you do with the user's account?
3. Which group scope can be assigned permissions in any domain or forest?

Common Issues Related to Group Management

Issue	Troubleshooting tip
Cannot convert group scope	
Cannot add group to another group	
Cannot create group in AD DS	

Real-World Issues and Scenarios

- A project manager in your department is starting a group project that will continue for the next year. Several users from your department and other departments will be dedicated to the project during this time. The project team must have access to the same shared resources. The project manager must be able to manage the user accounts and group accounts in AD DS. However, you do not want to give the project manager permission to manage anything else in AD DS. What is the best way to do this?

Best Practices for Group Management

- When managing access to resources, try to use both rule and role groups.
- Use Universal groups only when necessary because they add weight to replication traffic.
- Use Windows PowerShell with Active Directory Module for batch jobs on groups.
- Avoid adding users to Built-in and Default Groups.

Tools

Tool	Use	Where to find it
Active Directory Users and Computers	• Manage groups	Administrative Tools
Windows Power Shell with Active Directory Module	• Manage groups	Installed as Windows Feature
DS utilities	• Manage groups	Command line

Windows Server 2008 R2 Features Introduced in this Module

Feature	Description
Windows PowerShell with Active Directory Module	New administration utility for Active Directory, based on Windows PowerShell